RHODES KEYBOARDS INSTRUMENTS

CHROMA COMPUTER INTERFACE

SEQUENCER MANUAL

Tony Williams

DECEMBER 10, 1982

51 West 52 Street
New York, NY 10019

REV 4 ADDENDUM

This interface kit package contains software that has been slightly modified to accomodate the Apple IIE. Also included in the software is a utility program that makes interfacing to drum machines a little easier. Transposition and tuning functions have been added to the **H - CHROMA<>EXPANDER** menu selection. This Addendum to the Chroma Sequencer Manual describes these changes and additions, calls attention to a few errors found in the original manual, and gives more information on syncing to drum machines.

The Apple IIE

The Apple IIE was designed to be compatible with most software written for the II Plus. The IIE, however, uses zero page location $1F, which is not used by the II Plus and is used by our REV 3 software. Our software was, therefore, rewritten to use locations $03F5 and $03F6 instead of $1E and $1F. This new software, REV 4, will work with both II Plus' and IIE's.

If you write any software that uses the ampersand feature of BASIC, this software will not work, since $03F5 and $03F6 are the ampersand vectors.

If you are using a IIE, the connector box cannot be mounted to the computer. You must, however, provide a ground connection between the box and the computer. A good place to connect a wire on the box is the round cable strain relief. Any exposed metal on the computer will suffice as the other connection point.

Apple Computer, Inc. Makes no Warranties

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

Interfacing to Drum Machines

Also included in the REV 4 software is a User Utility program that simplifies interfacing to some electronic drum machines. The utility is automatically invoked after recording a track to rearrange time values for gated external clocks.

The interface hardware and software can sync as a slave to an external device. The external device (drum machine, sequencer, etc.) is a master that sends a clock signal to the Sequencer. The Sequencer then records and plays back according to the speed of the clock signal. The signal specification is detailed in Appendix O, however, an additional requirement is that the signal must be gated. In other words, the clock signal must not change until the external device begins to play and must stop changing when the device stops. Notice that the waveform must have a duty cycle between 25% and 75%.

The interface hardware clocks an external signal on the falling edge. Under most conditions this presents no problems when interfacing to positive edge devices that output normal clock frequencies. If the small delay (25 milliseconds maximum with a 24 pulse per beat clock at 50 BPM) is objectionable, you will have to invert the clock signal with hardware.

Previously, the musician was required to manually insert a time zero command at the beginning of the sequence. This operation was to be performed only after recording the first track, as detailed in the last paragraph of Appendix F. REV 4 software includes a User Utility program (User Bank 1, number 2) that performs the insertion for you. This utility is automatically called after each record operation, but it changes the sequence only once after the first track is laid down.

To record a sequence, the following steps should be followed:

(1) Connect a cable between the clock output of the external device to the EXTCLK input of the interface connector box.

(2) Use RECONFIGURE to select an external clock in the Sequencer software (see Chapter 9).

(3) Select a pattern on the drum machine with which to record. This will be your "click track." Make sure that the Sequencer's click track says NONE. The drum machine pattern can be edited later and still be in sync.

(4) With the drum machine not playing, go through the normal recording selections. Go all the way, until the inverse video message appears (RECORDING...).

(5) Start the drum machine playing. Play the Chroma along with the drum pattern whenever you want the music to start. Keep in mind that the time delay between the first drum beat and when you first play a note will be remembered and played back exactly.

(6) Stop the recording as normal, then stop the drum machine.

(7) Save the sequence, if desired, then repeat steps 4 through 6 until you are through with sequence.

To play a sequence the following steps should be followed:

(1) With the drum machine not playing, press **P** for PLAY or **W** for PLAY ALONG.

(2) When the Sequencer displays "PLAYING...", turn on the drum machine.

(3) Play to the end of the sequence or press the space bar or footswitch to stop.

(4) Stop the drum machine.

When recording or playing back, you must make sure that the speed change function is off (see Chapter 6). Also, some drum machines have start/stop switches that bounce if not depressed deliberately. If this happens, the two units will get out of sync because the drum machine will output a few pulses, then stop and start over.

With some drum machines, it is possible to create a song that stops at the end rather than looping forever. If this is the case, you should make the drum machine song end at the same time as the sequence. Full use of rests and special rhythm patterns at the beginning and end of the drum song will allow a professional sounding result.

Once the sequence is saved, the timer source (INTERNAL or EXTERNAL) is saved and reloaded along with the other data. In other words, it is not neccessary to change the timer source in RECONFIGURE when loading an externally synced sequence from disk.

It is not possible to record a sequence using the internal clock, then change to an external clock and have them be in sync. It is possible, however, to change the other way (external to internal) by changing the TIMINC and using the speed changing functions to increase the number of clock pulses required between events (the internal clock oscillates at 1000 Hz.).

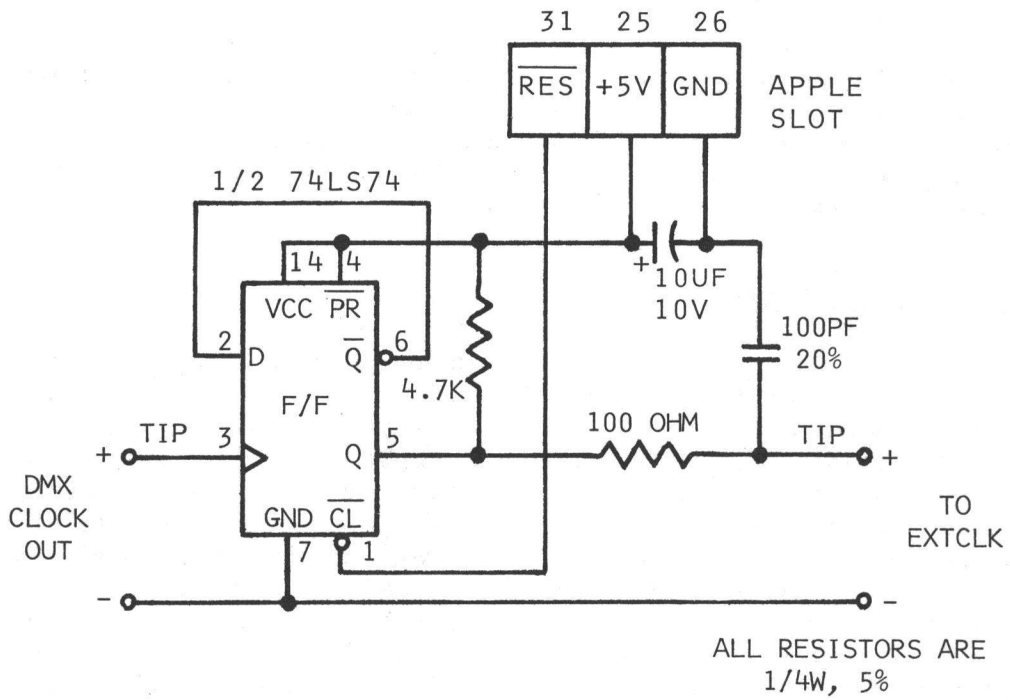
Looping works exactly like looping without a click track, meaning that it is very difficult. The time delay (count off) introduced when recording the first track is included in the loop. You must be careful when you stop the recording of the first track or any extended track, since this is also included in the loop time calculation. Modifying the timing with the editor (see Chapters 6 and 8) is possible and probably necessary.

Compatible drum machines currently consist of the LM1 Drum Machine, the LinnDrum (1) and the Drumulator (2). Connection between the LM1 and the Sequencer is between the Trigger output (set at maximum frequency) and the EXTCLK input of the Sequencer. Connection to the LinnDrum is between the SYNC OUT and the EXTCLK input of the Sequencer. The exact name of the signal on the Drumulator is unavailable at this time; but, it too is connected to the EXTCLK input of the Sequencer.

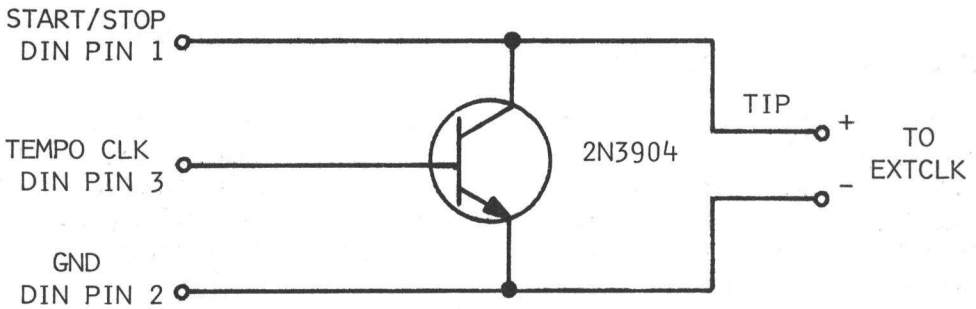
Other drum machines are not directly compatible. Small modifications to the interface hardware may allow you to sync them to the Sequencer software. These modifications are described below. However, <u>you will void your warranty if they are performed on the actual PC board</u>.

The Oberheim DMX (3) clock output has a constant pulse width of 104 microseconds. This is definitely less than the minimum required duty cycle of 25%. It is possible to change R48 on the interface PC card to allow this small pulse width. This method is not recommended since it may work differently for various clock frequencies and will void your warranty.

A better method is to route the clock signal to a toggle flip/flop (74LS74). This will provide a square wave at one-half the original frequency, which can be directly sent to the connector box EXTCLK input. The frequency division presents no problems, since the DMX is reasonably fast to begin with. The required IC should be mounted on an Apple prototype board and inserted into your computer. Power is available from the Apple slot. The following circuit should work (although not tested):



The Roland TR808 and TR606 (4) present a more complex problem, but an easier solution. The clock signal is not gated. Instead, the clock is free running and a start/stop signal is provided. It is possible to gate the free running clock with the start/stop signal, providing a compatible clock signal. The following circuit should work (although not tested):



The sync switch on the drum machine should be set to OUT since it will be the master. Pins 4 and 5 of the DIN connector are not used. This circuit could be mounted inside a custom-built cable. Insert 10K resistors in series with pins 1 and 3 if you intend to connect other devices to the drum machine outputs.

Help your fellow musician - if anyone tries these circuits, we would be interested in hearing from you. Please call Tony Williams at (617) 938-1610.

New Appendix G

Included in this manual is a new Appendix G, which describes how someone might use the existing assembly language routines in their own BASIC program. If you are seriously planning to write your own BASIC program, you should consider the I/O Driver described below.

I/O Driver Available

Many people have expressed an interest in writing their own software for the existing hardware. The Sequencer software package contains a lot of code that may not be used, thereby wasting valuable memory space. Most people, however, do not want to bother with writing all of the required low-level assembly language controlling software. In an attempt to accommodate our customers, we have written an I/O Driver that is available only to registered owners of interface kits. A software license agreement is required to protect our development investment. The software and full documentation will be licensed to you (for your personal use only) at a fee of $50. Please address your inquiries to: Fender Musical Instruments, 86 Cummings Park, Woburn, MA 01801.

Mistake in Schematic Diagram

The following mistake was discovered in the schematic diagram of the interface hardware: capacitors C58 and C57, shown connected to pins 4 and 5 of Z10, do not exist.

Changing the AUTO-SAVE Default

Presently, the AUTO-SAVE function is ON when powered-up. You can change this by escaping to FP BASIC (**<ESC>**), typing **105 AS%=255 <RET>**, then typing **SAVE RECORD TRACK.1,D1 <RET>**. Keep in mind that the modification should be done on a backup, not the original and that any write protect tab should be removed prior to attempting to save the program.

Arpeggiation in the Chroma

Several people have asked if the arpeggiator in the Chroma can be synced to a drum machine. The answer is no. The Chroma does not have an accurate hardware timer with which it references the arpeggiation speed. It establishes the speed by a software loop, which shortens or lengthens, depending on how busy it is. Although it is easy to record an arpeggio with the Sequencer, the exact number of notes heard will be different from when it was played. It will again be different after recording another track, because the Chroma will see an increase in the amount of data transmitted across the interface.

Error In Appendix B

There is an error in the first paragraph of APPENDIX B. Sequences are stored as "NEW TYPE B" files and program groups are stored as "NEW TYPE A" files.

Bugs Fixed in REV 4

There was a bug in REV 3 and REV 3E software that muted a track every time the sequence looped if the endpoint was set at anything other than the ending of the sequence. This problem was recently brought to our attention by a customer and we have incorporated the solution in REV 4.

If you discover an anomaly which you think is a software bug, please contact Fender Musical Instruments, 86 Cummings Park, Woburn, MA 01801 (617) 938-1610.

The occasionally obscure error messages in the Interface Test Program have been corrected.

The **H - CHROMA<>EXPANDER** selection now allows an Expander autotune without giving you an error message and returning to the main menu.

All functions relating to the new upcoming Chroma software revision (REV 13, or Interface REV 3) have been debugged, including pressure operations.

In the previous revision, a LOAD SETUP operation in RECONFIGURE set the disk slot to 7. If you had the disk controller in another slot, the system would hang on disk accesses. This software finds the slot number after loading a SETUP, so the problem is solved.

In REV 3E software, a recently released temporary update for Apple IIE owners, the User Utility Bank 1, Number 2 program crashed your sequence if you had a bad EXTCLK cable (no pulses received). The new software does not.

In the older software, the Edit Track List was always cleared when entering the Editor. The new software saves the list, so it only has to be set up once.

New H Stuff

REV 4 software has many added features in the Chroma to Expander communication routine (**H - CHROMA<>EXPANDER**). Not only does it allow the Chroma to send data to the Expander as if a cable connected them, it can also transpose the note and pressure information. There is also a new routine that copies the master tune value from the Chroma to the Expander port, which gives at least a good starting point on getting the two instruments in tune.

When **H** is selected, the following menu appears:

```
THE CHROMA AND EXPANDER CAN NOW TALK!


     T - TRANSPOSE  0
     U - UP 1 OCT
     D - DOWN 1 OCT
     0 - CLEAR TRANSPOSE
     1 - 0
     2 - 0
     3 - 0
     P - PROGRAM MODE
     C - COPY TUNING

  <ESC> - EXIT TO MAIN MENU
```

The zeroes are in inverse video and the computer is waiting for your selection.

COPY TUNING peeks at the two byte tuning value in the instrument on the Chroma port and pokes them into the instrument on the Expander port. In most cases, this tunes the two instruments together. You may have to fine tune the Expander.

TRANSPOSE allows you to set a transposition value between -33 and +33. When the value is inputted (follow the instructions presented when you type **T**) the inverse video field next to TRANSPOSE is updated. All notes in the Expander are squelched and any future attacks, releases and pressure commands are offsetted by the transposition value.

UP 1 OCT sets the transposition value to +12 and updates the inverse video field near TRANSPOSE.

DOWN 1 OCT sets the transposition value to -12 and updates the inverse video field near TRANSPOSE.

CLEAR TRANSPOSE sets the transposition value to 0 and updates the inverse video field near TRANSPOSE.

1, 2, and 3 are programmable preset transposition values that can be selected. The transposition value and the inverse video field near TRANSPOSE is set according the value shown in inverse video near the selection (1, 2, or 3). These values are retained, even if you exit to the main menu and come back to **H** - CHROMA<>EXPANDER. They are set to zero when the Sequencer is turned on.

PROGRAM MODE is entered to set the above mentioned programmable transposition presets. When **P** is selected, the computer will ask for a preset number, then a transposition value. If you type <RET> when asked for the preset number, the PROGRAM MODE will be aborted. The inverse video field near the preset number will be updated.

Trademark and Disclaimer Notices

(1) LinnDrum and LM-1 Drum Computer are registered trademarks of Linn Electronics, Inc.

(2) Drumulator is a registered trademark of E-mu Systems, Inc.

(3) DMX is a registered trademark of Oberheim Electronics, Inc.

(4) TR-606 Drumatix and TR-808 Rhythm Composer are registered trademarks of Roland Corporation.

Rhodes and Chroma are registered trademarks of CBS INC.

Apple II Plus and Apple IIE are registered trademarks of Apple Computer, Inc. APPLE COMPUTER, INC., was not in any way involved in the writing or preparation of this document or product, nor were the facts presented in the package reviewed for accuracy by that company. Use of the term APPLE should not be construed to represent any endorsement, official or otherwise, by APPLE COMPUTER, INC.

Notification that certain drum or rhythm machines are compatible with CBS' products should not be construed in any way to represent any endorsement, official or otherwise, by the respective rhythm machine manufacturers or by CBS INC.

CBS INC., its distributor, or its retailer is not responsible for any damage that may occur to your computer, its peripherals, or your drum machine, regardless of the cause or reason.

CONTENTS

1. INTRODUCTION
   A. General Description
   B. Warranty/Trademark/Copyright Information

2. INSTALLATION
   A. Required Equipment
   B. Installing the Board and Connector Box
   C. Power-up Procedure

3. MAIN MENU
   A. Basic Description
   B. Quick Reference To Selections

4. PLAYING SEQUENCES
   A. Loading From Disk
   B. Play/Play Along
   C. Stopping
   D. Looping
   E. Endpoints

5. RECORDING SEQUENCES
   A. Clearing a Sequence
   B. Setting up the Click Track
   C. Auto Save
   D. Recording a Clean Slate
   E. Recording Subsequent Tracks
   F. Recording at Slower Speeds
   G. Recording With Pressure
   H. Loop Recording

6. EDITING SEQUENCES
   A. Misc. Track Directory Functions
      1. Track Directory
      2. Change Port
      3. Change Program
      4. Change Volume
      5. Zap a Track
      6. Rename a Track
      7. Transpose
      8. Mute/Unmute
   B. Speed Changing
   C. The Editor

7. DISK FUNCTIONS
   A. Program File Management
   B. Disk Catalog
   C. Delete a Sequence
   D. Save a Sequence
   E. Load a Sequence

8. THE CLICK TRACK
   A. How It Is Generated
   B. The Need for a Click Track
   C. Time Signature
   D. How To Set It Up
   E. Restrictions on Changing the Click Track
   F. Measure Commands
   G. Loop Time

9. RECONFIGURE
   A. Description of the System Status
   B. Changing The Configuration
   C. Saving Set-ups
   D. Loading Set-ups
   E. Deleting Set-ups

10. MISCELLANEOUS
   A. Transfer (Chroma<>Expander)
   B. Reinitialize

APPENDICES

APPENDIX A. Error Conditions and Codes

APPENDIX B. Sequence/Program Data Structure

APPENDIX C. Overlay Structure and User Utility Programs

APPENDIX D. System Memory Usage

APPENDIX E. Getting More Notes Out of the System

APPENDIX F. Interfacing to Drum Machines and Other Sequencers

APPENDIX G. Slot Independency - Setting Up For Other Slots

APPENDIX H. Tracks, Channels, Boards and Instruments

APPENDIX I. Chroma Command Set

APPENDIX J. I/O Locations and Their Functions

APPENDIX K. A Software Bug in the Chroma Described

APPENDIX L. Interface Test Program (USER20)

APPENDIX M. Copying Diskettes and Creating New Sequence Data Disks

APPENDIX N. Using Other Systems While This Card Is In Place

APPENDIX O. Hardware Description and Specifications

APPENDIX P. Name Syntax Rules

CHROMA COMPUTER INTERFACE KIT

The Chroma Computer Interface kit contains everything you need to connect a Chroma with an Apple II. When installed, the Chroma can send and receive live performance information and programming information. A multitrack sequencer software package is included to get you started. Other utility programs, such as voice program editing and file management will soon be available, or you can write your own software utilities. With the Chroma's versatile command oriented communications language, the Chroma is the first intelligent music terminal; now you can create your own computer applications.

COMPUTER INTERFACE BUS

Anything that you play, select, or move on the Chroma and its accessories can be sent to a computer or other device over the Chroma Interface Bus (CIB). The external computer can then store the data and send it back to the Chroma in the form of a digital recording or sequence. Other applications of the CIB are voice programming, voice data storage, composition, transcription and production of film and video scores. As you can see, the Chroma and its interface bus are useful for studio as well as live performance and educational applications.

CHROMA COMMAND SET

The Chroma is structured around its interface bus - not just modified to accomodate it. It has available eight (8) 'Instruments', each of which may be programmed and played individually by the external computer. The interface command set is extremely versitile, including commands such as ATTACK, RELEASE, PEDAL 1, PEDAL 2, PITCH BEND, MODULATION LEVER, LATCH, SUSTAIN, VOLUME, PRESSURE, SET VOICE PARAMETER and others for each Instrument. Also included are commands for saving and loading voices, saving and loading packets of data from the cassette interface, peeking and poking into the Chroma's internal memory, and reading or changing the values of any program in CMOS memory. Currently, there are a total of 166 such commands and room for expansion to 255.

SEQUENCER SOFTWARE AND HARDWARE

Fender/Rogers/Rhodes now has available hardware and software to allow a Chroma and an Apple II computer to talk to each other. Included with the package are a set of application programs (in Applesoft and Assembly Language) that perform sequencing, editing and Chroma program data storage functions. The Sequencer can record up to 16 independent tracks and can simultaneously control a Chroma and the recently developed Expander. The Sequencer is fully polyphonic (up to eight (8) dual channel boards) and records key velocity and pressure as well as ALL performance controls and voice changes.

INTERFACE P.C. CARD

The printed circuit (PC) card hardware and the Interface I/O Driver software (included on the 5-1/4 diskette with the System software) are required to establish communication between the Chroma and Apple. The hardware is configured as two interrupt driven unidirectional 8 bit parallel ports and four control lines. The expander is controlled by 2 polled ports. The PC card also incorporates a one millisecond timing pulse for sequence timing and an Analog-to-Digital converter for pedal input of speed changes, etc. There are other single bit inputs for a footswitch and a sync pulse (for connecting to drum synthesizers and other sequencers, etc.). There is also a filtered pulse output for a audio click track. There are hardware and software provisions for easily hooking up an external clock. And finally, there are extra read and write pulses available if you want to modify the included hardware for your own software.

SEQUENCER SOFTWARE

The Interface I/O Driver software performs all the functions necessary to get data to and from the Chroma. The data is stored in two 255 byte FIFO (first in first out) buffers. Data coming from the Chroma gets two bytes of timing data as it arrives. All that is required to receive and transmit data is a JSR from assembly or a PEEK/POKE then CALL from BASIC. An assembly listing of the I/O Driver is included in the Programming Manual.

The Sequencer software is also included on the System diskette. This software receives data from the Chroma using the Interface I/O Driver described above and formats it for storage on the Sequence diskette in Drive 1 or 2. To record a track, all you really need to do is press 'R,' select a voice on the Chroma, press a footswitch and start playing. The Sequencer will remember all notes with velocities and pressures, all lever and pedal movements, all footswitch depressions, all voice changes, and even movements of the parameter control slider. When recording subsequent tracks, you need not start at the beginning of the sequence since the track starts wherever you play the first note.

The Sequencer will automatically store measures for endpoint and Editor references. The Sequencer has a built-in click track that can emphasize the first beat of the measure and has a continuously variable speed of 39-234 BPM.

To play a sequence all that is required is to press 'P' but there are many other options if you like - such as playing along, looping between two endpoints (defined by measures), muting tracks, etc. You can set a mode such that the Sequencer waits for a pulse (or footswitch depression) on the SYNC input before it starts playing.

Sequence editing can also be performed by changing the speed of the sequence, changing voices and volumes of individual tracks, changing a track from Chroma to Expander, deleting or changing notes, velocities and performance controls, transposing tracks, etc.

Finally, there are many functions that allow you to save sequences and groups of Chroma program data on diskette, display catalogs and set up direct communication between a Chroma and an Expander.

There are two banks of User Utility Program space if you want to write your own utilities. Each bank can contain up to ten (10) 2048 byte assembly language programs. We have included a few to get you started: a complete hardware test program, a program that allows you to write a 115 character comment in the sequence for your own use, a program that enables you to check the adjustment of the pedal conversion range, and an ECHO program that makes the Chroma sound like it is connected to a strange tremelo or delay device.

APPLE REQUIREMENTS

The Chroma/Apple system requires an Apple II Plus 3.3 (with 48K and Autostart Rom), a 16K RAM card (Microsoft or Apple Language Card or other compatible ram card), one or two disk drives (although two is recommended), a CRT monitor, the Interface Kit and, of course, a Chroma. If you want to control both a Chroma and Expander or two Chromas, the optional extra cable is required. The control pedal for speed changing is also option equipment (you can use your Chroma pedal if you want).

NOTE: The system will work if you have the old Monitor ROM only if you have an Apple Language Card or other compatible RAM card that provides the Autostart ROM.

WHAT IS IN THE KIT

Included in the Chroma Computer Interface Kit is the following:

- The Printed Circuit Card
- The Connector Chassis Assembly
- An Interface Cable for one instrument
- Sequencer Program Diskette
- Sequencer Data Diskette for storing your sequences (included are some sample sequences and program groups, including the 3 factory sets)
- Footswitch
- User's Manual

Optional accessories include:

- An extra cable to control an Expander
- A Control Pedal for speed changing
- A Programmers Manual (for those that want to write custom software for their Apple II computer)
- The Chroma Computer Interface Manual (discusses the Chroma Interface Bus with regard to computers in general)

SEQUENCE MEMORY CAPACITY: 28092 bytes or approximately 1860 notes. Actual note capacity depends on the amount of performance control changes and your playing style. You can get an extra 100 notes or so by using the Scrunch utility.

WARRANTY AND TRADEMARK INFORMATION

The words APPLE, APPLESOFT and APPLE LANGUAGE CARD are registered trademarks of APPLE COMPUTER, INC.

APPLE COMPUTER, INC. was not in any way involved in the writing or preparation of this document or product, nor were the facts presented in the package reviewed for accuracy by that company. Use of the term APPLE should not be construed to represent any endorsement, official or otherwise, by APPLE COMPUTER, INC.

MICROSOFT RAM CARD is a trademark of Microsoft Consumer Products, a division of Microsoft, Inc.

The hardware in this product is covered under the normal limited warranty of Rhodes/Chroma products.

CBS INC., its distributor, or its retailer is not responsible for any damage that may occur to your APPLE II computer, regardless of the cause or reason.

CBS INC. makes no warranties, either expressed or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. This software package is sold "as is". The entire risk as to its quality and performance is with the buyer. Should the programs prove defective following their purchase, the buyer (and not CBS INC., its ditributor, or its retailer) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages resulting from any defect in the software, even if CBS INC. has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so this paragraph may not apply to you.

ALL SPECIFICATIONS DETAILED IN THIS MANUAL ARE SUBJECT TO CHANGE AT ANY TIME WITH OR WITHOUT NOTIFICATION FROM CBS INC.

The Sequencer program and this manual were written December 10, 1982 by Tony Williams.

51 West 52 Street
New York, NY 10019

INSTALLATION OF THE SEQUENCER KIT

EQUIPMENT REQUIREMENTS:

To put the system together you need the following:

-one Chroma, a Chroma and Expander, or two Chromas.

-an Apple II Plus computer with 3.3 DOS with a 16K RAM Card (Microsoft RAM card or Apple Language Card or other compatible RAM card), one or two disk drives (although two drives is recommended), and a CRT monitor.

-a Chroma Computer Interface Kit, which consists of the printed circuit (PC) card, the interface cable, the phono jack box and mounting plate, footswitch, Sequencer Program diskette, Sequencer Data diskette, and the Sequencer Manual.

-options include another interface cable for controlling more than one Chroma, a control pedal for speed changing and a Sequencer Programming Manual, which tells you how to write extra software for the Sequencer system. A Chroma Computer Interface Manual is also available, which describes interfacing to the Chroma in general.

HOW TO INSTALL IT:

The following installation steps should first be read then performed in succession. If you need to install the PC card in a slot other than 5 see Appendix G.

1. Turn the power **off** both the Apple II and the Chroma(s).

2. Open the top cover of the Apple II.

3. Take the connector box with the mounting plate loosely attached and slip it onto the back of the Apple. Orient the box such that the cables slide in the second cutout from the left and the screws in the mounting plate are in the leftmost cutout. Tighten the screws in the mounting plate firmly (you may want to remove your RAM card from slot 0 to do this).

4. Orient the Interface PC card as it will fit into **slot 5** of the Apple PC board. Before actually inserting it in the slot, connect the round phono cable assembly from the box onto its wafer pin mate on the Interface PC card. Match pin 1 on the cable assembly connector to the **top** pin of the wafer connector. When it is properly connected, the metal pins in the connector will be visible from the component side of the PC board.

5. Next connect the ribbon cable from the Chroma port to the **top** 26 pin connector on the Interface PC card. The Chroma port cable is the leftmost ribbon coming out of the slot in the connector box. Then connect the Expander port ribbon cable to the **bottom** 26 pin connector on the Interface PC card. Then finally, slowly insert the PC card in **slot 5** of the Apple II.

6. Since it is not recommended that you have unnecessary current drawn from the Apple II supply, you should remove any peripheral cards that you will not need with this system. You will, of course, need the RAM card and the disk controller card. See the manuals for each of these boards regarding installation procedures.

7. Replace the Apple II top cover.

8. Install the large black external computer cable between the **COMPUTER INTERFACE** connector on the Chroma to the **CHROMA** connector on the box. It does not matter which end goes to the Chroma. Do the same for the Expander cable if you have one.

9. Plug the footswitch jack into the plug of the connector box marked "FOOTSW" and the control pedal jack into the plug marked "PEDAL" (if you have one). If you want to use the system without a footswitch see the RECONFIGURE section of this manual.

10. Connect your own 1/4" phone audio cable between the **CLICK** jack on the connector box and a high impedance input to an amplifier or mixer.

11. Follow the power up procedure on the next page.

POWER UP PROCEDURE:

1. Turn on the Chroma (and Expander if used).

2. Insert the Sequencer Program diskette in Drive 1 of the Apple II. If you are using two drives, insert the Sequencer Data diskette in Drive 2. If you are using only one drive, see Chapter 9, RECONFIGURE before you do any disk operations.

3. After the Chroma Auto Tune LED flash stops, turn on the Apple II and the CRT monitor.

4. A message will appear indicating that the system is initializing. Once it has determined that there is a Chroma connected, it will print "Chroma is on line...".

5. If the Chroma is not responding, the Apple II will tell you so, in which case you should check the cabling, reset the Chroma (set-split 50) then type any key on the Apple II keyboard. If this does not fix the problem then power down the Apple II, reset the Chroma, then turn on the Apple II again. If you are still having problems, follow the steps outlined in this chapter from scratch. If all else fails call the Rhodes Service Department at (617) 938-1610.

THE MAIN MENU

This chapter describes the main menu. At the top is the Memory Usage, which is an approximate percentage of the total memory that is used by the sequence currently residing in RAM. If there is an inverse video space (square white spot) after "G-GET SEQ" then there is a sequence currently in RAM. When the system is first powered up the Memory Usage is 0% and there is no inverse space at GEQ SEQ because a sequence has not yet been loaded.

There are three pages of main menu selections. Each of the entries represent functions that can be selected by pressing the appropriate key on the APPLE keyboard. These are single key entry selections; in other words, a <RET> should not be depressed after the letter. Any of the selections on all three pages can be accessed no matter which page is being displayed. The "^" symbol means that the "CONTROL" or "CTRL" key must be held down **at the same time** as the regular key. For example to select the speed change mode (^T-SPEED CHANGE), you would hold the CTRL key down and press T on the APPLE keyboard.

Some of the selections from the main menu invoke other menus or questions from the Sequencer. Other selections are switch functions, so pressing their key toggles a condition. Several functions have no meaning if there is not a sequence in memory (such as P-PLAY SEQ) so pressing that key does nothing and returns to the main menu. If you press two keys, one right after the other (not counting CNTL operations), the sequencer will remember the second key and apply it as the next selection. This "type ahead" feature allows fast key entry but you must be a little more careful when you make a selection. Do not be afraid to experiment since anything that can destroy data directly from the main menu has second chance features built-in.

The first page of the menu has special significance because it contains important information regarding the status of the various functions. As mentioned before, the GET SEQ function has an inverse video space if there is a sequence in RAM. Another example is the inverse video section after "K-Click Track" which tells the user the status of the click track. The status of the function displayed on page one of the main menu is described in detail in the sections of this manual dealing with those particular functions. After any complete sequence of operations are performed, page one of the main menu will be displayed.

If the CRT screen blanks all of a sudden while one page one of the main menu, there is nothing wrong; the Sequencer is just blanking the display so CRT damage will not occur if the system is left unattended for an extended period of time. To restore the display, hit any key. This key will not be decoded as a menu selection.

MENU SELECTION QUICK REFERENCE

G - GET SEQUENCE from diskette.

S - SAVE SEQUENCE onto diskette.

P - PLAY SEQUENCE (plays unmuted tracks).

W - PLAY ALONG (same as PLAY SEQUENCE but reserves instrument 0 and 1 (if link) for the Chroma keyboard).

N - SET ENDPOINTS (measure) for PLAY SEQUENCE, RECORD TRACK, in or out of LOOP mode.

R - RECORD TRACK (records a track, two tracks if program has a link).

A - AUTO SAVE SWITCH (if ON then sequencer automatically saves the sequence, with a rename option, at the end of each RECORD TRACK operation).

T - TRACK DIRECTORY (displays recorded tracks and provides useful information such as the name of the track, the port to which it is assigned (Chroma or Expander), its voice program number, and the initial volume of the track).

M - MUTE or UNMUTE a track or all tracks.

L - LOOP SWITCH (if ON then sequencer loops between ENDPOINTS. This allows PLAY ALONG during RECORD TRACK mode until cued for recording).

K - set up CLICK TRACK.

B - EDIT (allows user to do simple editing on a track(s)).

I - REINITIALIZE the Apple, the Chroma and the Expander, if present. This is used if any communication problem arises.

D - DELETE a sequence from the diskette.

E - CHANGE PROGRAM (voice) of a track(s).

C - CLEAR sequence from Apple memory.

F - CATALOG (displays a list of all sequences and program groups on the diskette).

Z - ZAP A TRACK (deletes a track(s) from the sequence in Apple memory).

Q - PROGRAM FILE MANAGEMENT. This allows you to transfer program groups to and from the Chroma, Expander and Disk. It also allows deletion of program groups from disk.

V - CHANGE VOLUME of a track(s) overall.

X - RECONFIGURE. This allows the user to disconnect the foot-switch, connect the SYNC INPUT, set up an external clock, change the slot locations of the Interface Card, or disable the emphasis on the first beat of the measure in the click track. It also allows the user to view the current status of the system.

H - CHROMA<>EXPANDER. Allows direct communication between the Chroma and the Expander ports.

Y - PRESSURE RECORD. Allows the Chroma port to send pressure information (works on REV 3+ Chromas with pressure sensor option only).

<RET> - Menu Page 1 <ESC> - EXIT to Applesoft.

2 - Menu Page 2 3 - Menu page 3

CNTL U - UTILITY BANK 1. Runs one of up to 10 user written program overlays (as an example we have included a comment program and an ECHO program).

CNTL V - UTILITY BANK 2. Runs one of up to 10 more user written program overlays, if available (as an example we have included a complete hardware test program).

CNTL X - delete all MEASURE commands.

CNTL Y - SCRUNCH the sequence (This reduces timing resolution of the sequence in memory and saves approximately 6% of the storage space).

CNTL T - Selects SPEED CHANGE mode. Selecting this toggles between OFF, SMALL, and LARGE time variations. The large speed change mode varies the playback speed of the sequence from 1/2 speed to 2 times normal speed continuously with the pedal. The small speed change mode does the same except within a more restricted range.

CNTL S - STORES the speed change as you here it if ON and in PLAY mode or OFF if in RECORD mode.

CNTL A - TRANSPOSE sequence or track.

CNTL N - RENAME a track(s).

CNTL P - CHANGE PORT from Chroma to Expander or vice versa.

PLAYING SEQUENCES

This section explains how to play one of the sample sequences provided with the system or one of your own sequences. It also goes into the details of looping and setting endpoints.

Before you load your first sequence it is a good idea to get familiar with the disk catalog selection. You should read about the disk catalog in Chapter 7 of this manual.

HOW TO LOAD A SEQUENCE:

This section tells you how to load any one of the sample sequences included on the data diskette supplied with the Interface Kit. This same procedure is to be used to load YOUR sequences once you have saved them onto disk.

When you are in the main menu, look at 'G-GET SEQ'. If there is an inverse video space after this menu entry then there is already a sequence in memory and it will be written over (lost) if you load another. See Chapter 7, Disk Functions, to learn how to save a sequence if you want to keep this one.

After typing 'G' for GET SEQ, the sequencer will ask you if you want to see the disk catalog. Type 'Y' if you do, any other key if you don't. After the catalog routine is done, as described in Chapter 7, the Sequencer will ask you for the name of the sequence to be loaded. If you change your mind at this point, just type <RET> and you will get back to page one of the main menu without loading the sequence. Otherwise, type in the name of the sequence then <RET>. If all goes well (meaning that the file was found, etc.), the Sequencer will tell you that the sequence was loaded and will display page one of the main menu.

PLAY / PLAY ALONG:

When playing a sequence you have a choice of "P-PLAY SEQ" or "W-PLAY ALONG". PLAY ALONG reserves instrument 0 for the Chroma keyboard, whereas PLAY SEQ defines the first track encountered as instrument 0. What this means is that under PLAY SEQ any lever changes, etc. that you do on the Chroma while the sequence is playing will be heard and any notes you play on the Chroma keyboard will use up channels that the first track would normally have available for its notes. Also, if you press a voice selection switch on the Chroma, that first track will change voices. In other words, the first track and what you do on the Chroma "shares" instrument 0. Under PLAY ALONG what you do on the Chroma is reserved for just that and the first track uses another instrument. The concept of tracks, instruments and channels is explained in Appendix H.

For now, just type 'P-PLAY SEQ' and you will hear the sequence that you have in RAM. If you are using the Sequencer in a live performance situation and want to start the sequence at an exact point in time, you should read Chapter 9, Reconfigure, regarding the use of the SYNC input with a footswitch.

NOTE: Some of the sample sequences contain tracks that are designated for output to the Expander port. If you play these sequences unaltered and there is no Expander on line, the Sequencer will tell you and automatically mute the track. To eliminate this problem, change the port and unmute the track(s) as described in Chapter 6. Then save the modified sequence as described in Chapter 7.

STOPPING THE SEQUENCE:

You can let the sequence play to the end or type a space or depress the footswitch to stop. Whichever you decide to do, the main menu will reappear with something on the top that says "LAST EVENT TIME: XXXXX". This is useful when editing a sequence so, for now, disregard it.

LOOPING:

What if you want the sequence to play over and over again? To do this type 'L' to toggle the LOOP SWITCH. Now if you play the sequence it will loop continuously until you press the space bar or tap the footswitch. When you stop the sequence the loop switch is automatically turned off. Whether the sequence loops in time depends on the way it was recorded and whether there was a click track when it was recorded. See the sections regarding RECORDING and the CLICK TRACK for more details about loop time.

ENDPOINTS:

What if you want to start the sequence at a point other than the beginning or end the sequence at a point other than the end? The Sequencer can use "measures" to define endpoints in the sequence. If the sequence was recorded with a click track, measures were automatically recorded in the sequence for reference. If the sequence you currently have in memory was recorded with a click track, the inverse video section after the selection "K-CLICK TRACK" on page one of the main menu will say "OFF 60BPM 4/4" or something similar. If it was recorded without a click track the inverse video section will say "NONE", in which case you cannot set up endpoints because there are no measures. If there was a click track, type 'N-ENDPOINTS' to set up the beginning and ending endpoints.

When you select ENDPOINTS, the Sequencer displays the total number of measures stored during the recording and the current endpoints. If the number of measures stored is zero then either the sequence was recorded without a click track or the person that recorded it deleted all the measures to save memory (see Appendix E, GETTING MORE NOTES OUT OF THE SYSTEM). If the current endpoints are "B-E" then the sequence would start playing at the beginning and end at the end, regardless of measures. To set the beginning endpoint type 1 then type the number of the measure you want to start at followed by a <RET>. To set the ending endpoint type 2 then type the number of the measure you want to stop at followed by a <RET>. If you try to change either endpoint to an illegal measure number (such as 0 or a number greater than the total measures stored) then no change in the endpoints will occur. Type a <RET> to get back to the main menu after setting the endpoints.

PLAYING AND LOOPING BETWEEN ENDPOINTS:

When you PLAY SEQ now, the sequencer will play, starting at the beginning measure and stopping at the ending measure set by the endpoints. If you now select LOOP mode by toggling the loop switch, the sequencer will play between the endpoints continuously until stopped. The loop time may be increased by one measure due to the increased processing time required to find the beginning endpoint.

RECORDING SEQUENCES

This section will explain the operations required to record a clean slate (the first track) and subsequent tracks. Before recording the first track of a sequence it is essential that the RAM be cleared. Therefore, this section will start with an explanation of C-CLEAR SEQUENCE.

C-CLEAR SEQ

Selecting this menu function initializes the Sequencer to begin recording. The Sequencer allows you to abort this command, as with all commands that will cause loss of valuable data. During this selection the RAM pointers are zeroed, various variables are initialized and the click track is turned off. The Memory Usage will obviously be zeroed and the G-GET SEQ menu entry will indicate that no sequence is in memory.

SETTING UP A CLICK TRACK

The click track is not really a "track" in a normal sense but a hardware/software output pulse that is controlled by the timer and various conditions set by the user. The information supplied by the musician is used for many other timing related operations such as loop time determination, measure command storage and endpoint references. It is therefore highly recommended that the user start recording a sequence with a click track. To do so you should read chapter 8, THE CLICK TRACK in this manual.

AUTO SAVE

Before you begin recording you should know about the AUTO SAVE function. If the AUTO SAVE SWITCH is ON, the Sequencer will enter the SAVE SEQ mode at the end of every record operation. This means that the Sequencer will display the current name of the sequence and ask if you want to rename it. If you want to abort the AUTO SAVE function, type <ESC> at this point (as in the case of the normal SAVE SEQ operation). If you do not type 'Y' (or <ESC>) the Sequencer will save the sequence at the current name. The AUTO SAVE function is useful for saving different versions of a sequence as you build it track by track. It is suggested that you leave the AUTO SAVE switch ON until you become familar with the Sequencer.

RECORDING A CLEAN SLATE

After the the RAM has been cleared, the click track set up and the AUTO SAVE switch is set at the desired state you are ready to record the first track. Type 'R' for RECORD TRACK and the Sequencer will ask you to name the sequence for future reference. Type <RET> if you do not want to be bothered with this but keep in mind that the sequence must be named before it can be saved (the sequencer will tell you this if there is no name and you try to save the sequence). Otherwise type in a name up to 15 characters long and press <RET>.

The Sequencer will then ask you to select a voice on the Chroma. Although you can change the voice of a track very easily after it is recorded, it is a good idea to select a voice that at least approximates what you will eventually want. This is because the sound of that voice will influence your phrasing and other aspects of the music. After you select a voice press any key to tell the computer you are ready.

The Sequencer will then ask you to name the track for **your** future reference. If you do not want to be bothered, just type <RET> and the name will default to 'Tn' where n is the track number. You can always change it later using the ^N-RENAME menu selection (more on this in Chapter 4, EDITING SEQUENCES).

After all the above is done, the Sequencer will display the tracks set up to record with and their names. It will then tell you to:

```
PRESS ANY KEY OR THE FOOTSWITCH TO BEGIN RECORDING...
PRESS ANY KEY OR THE FOOTSWITCH TO END RECORDING...
```

When you press either the footswitch or any key on the Apple keyboard, the inverse video **RECORDING...** message will be displayed. At this point the Sequencer is waiting for you to play. The track does not start and memory is **not** being used up until you start playing or you move a performance control.

Whenever a voice is changed during the record operation (by pressing a voice selection switch on the Chroma), you will create another track (two tracks if the voice you select has a link). The Sequencer will detect this, set up the track and, when you are through recording, ask you for the name of the track(s) that you created while recording.

The Sequencer can record up to 16 tracks but the Chroma can only play 8 tracks at a time. Changing voices while in the record mode is an easy way to concatenate tracks, insuring that there is no overlap. This allows you to use all 16 tracks even if you do not have an Expander or another Chroma.

When you are through recording the track(s), press any key on the Apple keyboard or depress the footswitch. If the Auto Save switch is on, then you will enter the SAVE SEQ mode as decribed above.

RECORDING SUBSEQUENT TRACKS

To record subsequent tracks you basically follow the same procedure as for a clean slate. The only difference is that the Sequencer will not ask you for the sequence name. You may want to turn off the Click Track, so see Chapter 8, THE CLICK TRACK. You may want to adjust the volume or mute certain tracks to enable you to hear the beat more easily (see Chapter 6, EDITING SEQUENCES). You may want to send certain tracks to the Expander so you have more notes available on the Chroma while recording the next tracks, etc.

RECORDING AT SLOWER SPEEDS

You may want to record at a slow speed to enable you to play difficult passages.

AT HALF SPEED:

If you record at half speed, the recording will double in speed on playback. This is very easy to do using one of two methods. The first involves using the speed changing pedal, but this should be done after you are completely through with the sequence since the click track will no longer be in sync with the music if you do it this way.

The second way is much better. This way involves the use of the RECONFIGURE selection described in Chapter 9. Basically, you want to set the TIMER INCREMENT to 1/2X before recording, then change it to 1X for playing. The final timer increment will be saved with the sequence and the click track will still be in sync. The only problem with this approach is that you can only change the timer increment in powers of two (1/16, 1/8, 1/4, 1/2, 1, 2, 4 and 8). As mentioned in Chapter 7, this feature was designed mainly to scale the external clock frequency.

AT OTHER SPEEDS:

You can record at any reduced speed using the SPEED CHANGE selection from the main menu, as long as you are willing to live without a click track. In record mode, the speed variation will be stored if the STORE SPEED switch is OFF. The reason for this is that the Sequencer determines the speed stored as the ratio of a variable to a fixed increment timer. You do not have to understand this to make the thing work.

RECORDING WITH PRESSURE

Since pressure recording requires a high data rate and uses a lot of memory, we have separated the record mode into two types: recording with pressure and recording without pressure. To record with pressure, select **Y - PRESSURE RECORD** and proceed as normal. If your Chroma does not have the pressure sensor option, you will not be wasting memory.

LOOP RECORDING

This method of recording essentially allows you to play along in loop mode, then easily tell the Sequencer when you have the passage down enough to record it. Once the Sequencer has been "cued", the next loop is in record mode.

First, turn ON the loop switch, then go through the normal record mode motions. When you press the space bar or footswitch to RECORD, the Sequencer will say "LOOPING...". Keep playing along with the sequence until you think you can record, then press the footswitch. The Sequencer will then say "CUED...". The next time the sequence loops, you will be in record mode and the Sequencer will say "RECORDING...". The sequence will stop and the record mode will be terminated when you press the footswitch again, press the space bar or the end of the sequence is reached.

EDITING SEQUENCES

This section deals with the many ways to edit your sequence once you have recorded it. Editing functions fall into three catagories: miscellaneous track directory functions, speed changing and the Editor.

MISCELLANEOUS TRACK DIRECTORY FUNCTIONS

These functions are grouped together because they all involve using and/or modifying the sequence track directory.

TRACK DIRECTORY

The track directory is a list of all the tracks that make up a sequence. Included in each track entry is the following vital information:

-track number- this is the number to use whenever you want to refer to a particular track.

-track name- this is currently for your use only. The Sequencer does not refer to this name except to check and see that it is unique among tracks in this sequence. The name follows the syntax rules outlined in APPENDIX P.

-track port- this is 'C' if the track is to be outputted to the Chroma port or 'X' if the track is to be outputted to the Expander port.

-track program number- this is the Chroma/Expander Program number for this track.

-track volume- this is the initial linear volume value for the track.

CHANGE PORT

This routine begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will toggle that track's port and redisplay the track directory. If the track was destined for the Chroma, it will now be set for the Expander, whether there is an Expander on line or not. During playback, if there is no Expander, the Sequencer will tell you and mute that track. If this happens, change the port back to Chroma and unmute the track as described later in this chapter. To exit the change port mode, press <RET> when the Sequencer asks you for the track number.

CHANGE PROGRAM

This routine also begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will prompt you for the new program number. If you change your mind and do not want to change this track, press <RET> and the sequencer will redisplay the track directory and ask for another track number to change. After you input the new program number, the Sequencer will change that track's program and redisplay the track directory. To exit the change program mode, press <RET> when the Sequencer asks you for the track number.

CHANGE VOLUME

This routine also begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will prompt you for the new initial volume value. If you change your mind and do not want to change this track, press <RET> and the Sequencer will redisplay the track directory and ask you for another track number. If you input the new initial volume value, the Sequencer will change the track volume and redisplay the track directory. To exit the change volume mode, press <RET> when the Sequencer asks for the track number.

The volume value is linear, as mentioned above in the description of the track directory. The volume of the track may be varied at any time from the initial value by use of a pedal controlling the volume parameter of a program in the Chroma during recording or by inserting volume commands via the Editor. Volume commands can also come from the Chroma when you are recording a linked program (2 tracks at once) and you change the link balance (see APPENDIX K for a description of a small software bug in the Chroma). If either of these is the case, then changing the initial volume of the track will only effect the track up until that volume command is encountered. In other words, the volume values, initial or otherwise, are absolute, not relative.

ZAP A TRACK

This routine will delete a track from the sequence. It begins by displaying a track directory and prompting you for the track number. After you input the track number, the Sequencer will delete all commands that belong to that track, delete all time commands that are followed by another time command then redisplay the track directory. To exit zap mode, press <RET> when the Sequencer asks you for the track number to delete.

If you zap the last track that was recorded, then that track number will be available for subsequent recording operations. If you zap a track other than the last one, that track number is not available for subsequent recordings. In other words, you effectively lose one of your 16 tracks and the tracks numbers will not be sequential.

RENAME A TRACK

This routine allows you to change the name of a track. It also begins by displaying the track directory and prompting you for the track number. After you input the track number, the Sequencer will prompt you for the new name. The name must follow the syntax rules as outlined in APPENDIX P, but don't worry, the Sequencer will not let you input anything that is illegal. If you change your mind and do not want to change the name of this track , press <RET> at this time. If you input the new name, the Sequencer will change the name of the track and redisplay the track directory. To exit the change name mode, press <RET> when the Sequencer asks you for the track number.

TRANSPOSE

This routine allows you to transpose the whole sequence or any track in semitones. Maximum allowable transposition is 33 semitones at any one time, up or down. The Sequencer currently does not check for cumulative transpositions, so care must be taken that the total amount of transposition does not exceed 33 semitones, else the Chroma will get confused and play the wrong notes. Unfortunately, after the limit has been passed, you cannot recover the sequence by transposing an equal amount in the opposite direction.

The routine begins by asking you if you want to transpose all tracks. If you do, type **A** for all. If you want to transpose one track, then type <RET>.

If you type <RET> for single track transposition, the Sequencer will display the track directory and prompt you for the track number. If you type <RET> instead of the track number at this point you will abort the transposition and exit back to the main menu. If you type a track number, the Sequencer will then ask you the amount to transpose. Type <RET> at this point to abort the transposition and exit back to the main menu, otherwise type in a semitone value preceeded by a + or - for direction. The + direction symbol is optional; in other words, if there is no direction symbol, the Sequencer will assume + direction. If you input an invalid value, the Sequencer will tell you and ask you to reenter the direction and transposition amount. After the transposition is complete, the Sequencer will return to the main menu.

If you type **A** to transpose all, as presented above, the Sequencer will ask you for the direction and amount of transposition. Enter this value as if you were transposing only one track as outlined above.

MUTE/UNMUTE

This routine allows you to mute and unmute tracks. A muted track retains all of its information, but will not be heard during playback. MUTE/UNMUTE has its own menu that allows you to view the track directory, mute a track, unmute a track, or unmute all tracks. A track that is muted will be shown in INVERSE VIDEO in the track directory (and the EDITOR) and will not be outputted to it's port. To exit the Mute/Unmute mode, press <RET> when the mute menu is displayed.

A track may be automatically muted if it is designated for the Expander port and no Expander is on line. It will also be automatically muted if you try to output more than 8 tracks (7 tracks if recording a non-linked program, 6 tracks if recording a linked program) to the Chroma port and the AUTO-REROUTE flag is off or there is no Expander on line. The muting ocurrs when the Sequencer encounters the start of a track that puts it over the limit. A message will be displayed and the Sequencer will continue to play/record. To permanently unmute a track that the Sequencer automatically mutes, you must correct the condition that caused this to happen. In other words, you must ZAP the track or change the port of the track.

SPEED CHANGING

You can change the speed of your sequence three different ways:

(1) By using a variable external clock (see APPENDIX F).

(2) By changing the time increment by powers of two (see Chapter 9, RECONFIGURE).

or (3) By using the main menu selections **^T - SPEED CHANGE** and **^S - STORE SPEED**, which is described in this section of the manual.

Before we begin, make sure you have a control pedal connected to the PEDAL input of the connector box on the back of your APPLE. You can use the Chroma's pedal, purchase one from us, or use any 100K linear taper potentiometer (it doesn't have to be a pedal).

The analog-to-digital converter (ADC) IC on the Interface PC Card will convert the resistance of your pedal into a digital value that the Sequencer will use to change the speed of the sequence. If you suspect that the range of the ADC is out of adjustment, run the ADC Test Program in USER UTILITY BANK 2, number 1 or the Interface Test Program in USER UTILITY BANK 2, number 0 (the command is ^A for ADC test in the Interface Test Program). Number 1 displays the value in decimal (0-255) while Number 0 displays it in HEX (00-FF). When you move the pedal to its extremes, the value displayed should go from 0 (pedal all the way up) to 255 or FF (pedal all the way down). If this is not the case, then you should adjust the ADC according to the procedure

outlined in APPENDIX O, Hardware Description and Specifications.

The speed change selection ^T is a three way toggle switch, which is either OFF, SM (small variation), or LRG (large variation). The large variation allows you to increase the speed up to twice normal or decrease it down to one-half normal. The small variation range is approximately one fourth of the large range. When either the small or large speed variation has been selected, the Sequencer will play back at a variable speed, depending on the position of the pedal.

In play mode, the speed change is not permanent unless the SPEED STORE switch is ON. In record mode, the speed change is not permanent unless the the SPEED STORE switch is OFF. This switch is a normal two position toggle switch. In the on position, the Sequencer changes the time values in the sequence to correspond to the variations in pedal position. When the switch is on, the sequence should be allowed to play all the way through, otherwise you will get timing glitches, or sudden changes in tempo, that are almost impossible to remove. If you want a sudden change in tempo, it is relatively easy to just move the pedal quickly.

A neat effect is to loop a small sequence with the SPEED CHANGE on large or small variation and the STORE SPEED on. Everytime the sequence loops, it is faster.

After the speed of a sequence has been permanently changed by STORE SPEED, the click will never again be in sync with the sequence. You should, therefore, change the speed of the sequence only after you no longer need the click track. The measure commands will be correct at the faster or slower speed because time values are stored with them, unlike the click track which is not really a track at all.

WARNING: Always turn off the SPEED STORE switch immediately after you permanently change the speed of the sequence. It does not automatically turn off. If you immediately play the new sequence to hear how it came out and the STORE SPEED switch is still on, you will change the speed again.

THE EDITOR

Each event in the sequence, such as Attacks, Releases, Performance Control changes, etc. are represented by commands stored in RAM. The Editor allows you to MONITOR, CHANGE, DELETE, or INSERT these commands.

The Editor operates by displaying the next command entry from the EDIT TRACK LIST and prompts you for instruction by a ":" on the next screen line. Command entries from muted tracks will be displayed in inverse video. If you do not want to change this command entry, just press the space bar (NEXT) and that command will be restored in the sequence. If the MONITOR is on and the track is not muted, then the command entry will also be outputted to its port. The next command entry will then be displayed. The sequence cannot go backwards. The endpoints are set at the beginning and ending of the sequence while in the Editor, regardless of previous endpoint settings. The endpoints will be restored upon EXIT from the Editor.

You can select from any of the instruction commands displayed at the top of the screen when the Sequencer is prompting you with a ":" symbol. Following is a detailed explanation of each instruction command:

CNTL M or <RET> - MONITOR on/off (allows you to hear what you are editing). If the MONITOR is on and the track is not muted, any command that is restored is also outputted to its port.

CNTL C - CHANGE currently displayed command entry. The Editor will redisplay the current command entry and position the cursor at the beginning of the command. At this point you can type in the full command entry as described below in INSERT or you can use the left and right arrow key to edit the line. Once <RET> is pressed, the Sequencer will erase the rest of the command entry and display the edited line, asking you if it is correct. If you type **Y** or <RET>, the Editor will restore and output the command entry, then display the next one. If you type **N** then the Editor will ask you to re-enter the command. You must type in the full command entry as described below in INSERT. Once this is done, the Editor will again display the edited line and ask for verification.

NOTE: The Editor will not assemble inverse video text, in other words, if you want to change a command entry that is muted, you must type in the full command as described in INSERT.

WARNING: Do not change a command entry to a BEG OF SEQUENCE or END OF SEQUENCE command or you will confuse the Editor and LOSE YOUR SEQUENCE.

WARNING: Do not change a command entry to a PRESSURE command if your Chroma and/or Expander is not Interface Software REV 3+. This will confuse the Editor and you will LOSE YOUR SEQUENCE. You can change to pressure commands, even if you do not have the pressure sensor option, as long as the software in your Chroma is REV 3+.

CNTL I - INSERT command before currently displayed command entry. Type in the full command entry as described below under EDITOR COMMAND ENTRY SYNTAX. Alternately, you can type in the shorthand version containing only the main command opcode followed by the vital numbers, separated by a space. The Editor will ask for verification as described above in CHANGE then store and output the command to its port (if not muted).

WARNING: Do not insert a command before a BEQ OF SEQUENCE command entry or you will confuse the Editor and LOSE YOUR SEQUENCE.

WARNING: Do not INSERT a PRESSURE command if your Chroma and/or Expander is not Interface Software REV 3+. This will confuse the Editor and you will LOSE YOUR SEQUENCE. You can INSERT pressure commands, even if you do not have the pressure sensor option, as long as the software in your Chroma is REV 3+.

CNTL D - DELETE currently displayed command entry. The Editor will display the current command and ask for verification. If you type **Y** or <RET> the Editor will not restore or output this command entry and will display the next one in line.

WARNING: Deleting MEASURE command entries other than the last one may cause errors in the ENDPOINT function if the deleted MEASURE command is specified as an ENDPOINT.

CNTL F - FIND command entry. The Editor will display the FIND: prompt at which point you have the following options:

(1) <RET> finds and displays the last command entry you asked the Editor to find.

(2) Typing a command entry as described above in INSERT then <RET> finds the command and displays it. Any section of the command left out is "wild", in other words the Editor doesn't care. For example, FIND: ATTACK 1 -20 will find the next ATTACK in TRACK 1 that has a key number of -20, regardless of the VELOCITY or PRESSURE.

NOTE: If the command entry is not found, then END OF SEQUENCE will be displayed. Type a space bar (NEXT) or ^H (HOME) to get to the BEG OF SEQUENCE before trying again.

NOTE: The Editor can find time values that span a time value "distance" of up to 32768 relative to the next time value. You must, therefore, make an intermediate "stop along the way" when finding time values greater than 32768 higher than the next time value.

<SPACE> - DISASSEMBLE and display NEXT command entry. This instruction will cause the current command entry to be restored and outputted to its port. The next command is then disassembled and displayed.

CNTL X - EXIT Editor. The Editor clears the EDIT TRACK LIST, secures the sequence pointers, restores the ENDPOINTS and exits back into page one of the main menu.

CNTL E - Add a track to the EDIT TRACK LIST. Type the TRACK number followed by a <RET>. If that track does not exist, the Editor will tell you that the entry is invalid and ask for another track number. Typing <RET> without a track number aborts this mode.

CNTL R - REMOVE a track from the EDIT TRACK LIST. Same as above, but removes a track from the list instead of adding it.

CNTL S - STATUS, displays the EDIT TRACK LIST. The EDIT TRACK LIST is a list of all tracks that are currently being edited. If a track is not being edited, the Editor will not display or output commands in that track. Also, you cannot INSERT or FIND command events in that track.

CNTL H - HOME returns to the beginning of the sequence.

NOTE: If an error occurs, the current instruction is aborted and an error message will be displayed. See APPENDIX A for an explanation of these messages.

EDITOR COMMAND ENTRY SYNTAX

The following is a list of the command entries that are allowed in the Editor and their correct syntax. An alternate shorthand syntax is allowed in most cases which consists of the command opcode (**BOLD** capitalized) followed by the parameters (numbers only), separated by a space.

Operations on DEFINE and UNDEFINE are limited to display only. Deleting a BEG OF SEQUENCE or END OF SEQUENCE command is not allowed. Unfortunately, the Editor does not prevent you from INSERTING or CHANGING to these commands, which will definitely cause lose of the sequence as described in the WARNINGS above.

When a number is specified to be in signed 2's complement form, it means that values from 0 to 127 are as normal and values from 128 to 255 are -128 to -1, respectively. In other words, the range is from 127 to -128 and the values -128 to -1 are represented by adding 256 to the negative number.

THE COMMAND ENTRIES:

BEG OF SEQUENCE

END OF SEQUENCE

TIME XXXXX where XXXXX is from 0 to 65535

MEASURE XXX where XXX is from 1 to 255

DEFINE TRACK#tt LEVER1=aaa LEVER2=bbb PEDAL1=ccc
PEDAL2=ddd VOLUME=eee FOOTSWITCH=fff
where tt is from 1 to 16
aaa, bbb, ccc, ddd and eee is from 0 to 255 (all except eee are in signed 2's complement form)
fff is 0 if both footswitches up
128 if LATCH down, SUSTAIN up
64 if LATCH up, SUSTAIN down
192 if both footswitches down

UNDEFINE TRACK#tt where tt is from 1 to 16

ATTACK TRACK#tt KEY#kkknn VELOCITY=vvv PRESSURE=ppp
where tt is from 1 to 16
kkk is from -64 to 63
nn is the musical note (this is not required when inputting, it is for your reference only)
vvv is from 0 to 31
ppp is from 0 to 63

RELEASE TRACK#tt KEY#kkknn VELOCITY
where tt is from 1 to 16
kkk is from -64 to 63
nn is the musical note
vvv is from 0 to 31

VOLUME TRACK#tt VALUE=vvv
where tt is from 1 to 16
vvv is from 0 to 255

LEVER1 TRACK#tt VALUE=vvv
where tt is from 1 to 16
vvv is from 0 to 255 in 2's complement form

LEVER2 TRACK#tt VALUE=vvv same as above

PEDAL1 TRACK#tt VALUE=vvv same as above

PEDAL2 TRACK#tt VALUE=vvv same as above

SUSTAIN UP TRACK#tt where tt is from 1 to 16

SUSTAIN DOWN TRACK#tt same as above

LATCH UP TRACK#tt same as above

LATCH DOWN TRACK#tt same as above

PRESSURE TRACK#tt KEY#kkknn VALUE=pp
where tt is from 1 to 16
kkk is from -64 to 63
nn is musical note
pp is from 0 to 63

SET PARAMETER TRACK#tt PARAMETER#ppp VALUE=vvv
where tt is from 1 to 16
ppp is from 1 to 100
vvv range depends on parameter (see Chroma Performance Manual)

EDITING TIPS:

When trying to find a note, it is advantageous to play the sequence and notice the LAST EVENT TIME: XXXXX message, then go into the Editor and find that time value. You must stop the playing of the sequence just prior to the note you are looking for.

When trying to add or subtract a time offset to every time command after a particular point in the sequence, use FIND:TIME then <RET>. After the first time value has been changed, use FIND: <RET> and the Editor will find the next time value and display it for changing. Do not be too ambitious with regard to changing the times of large sections of the sequence.

DISK FUNCTIONS

This section describes the Program File management, Sequencer catalog and saving, loading and deleting sequences.

PROGRAM FILE MANAGEMENT

Typing **Q** from the Main Menu displays a Program File Management Menu that allows you to transfer program data to and from the Chroma port and disk or from the Expander port and disk. Direct transfer of Program 0 between the Chroma and Expander port can be accomplished via the **H**-Chroma<>Expander main menu selection.

A Program Group is defined as a group of one or more 59 byte data blocks that make up the program. At present, the Sequencer only supports transfer of program groups containing 50 programs. The Program File Management Menu choices are as follows:

1-Chroma programs to disk
2-Disk programs to Chroma
3-Delete disk program group
And, if an Expander is on line,
4-Expander programs to disk
5-Disk programs to Expander

The delete menu selection (3) provides the user a second chance since valuable data can be lost. The disk catalog can easily be accessed when transferring from disk (2,5) and when deleting a file (3).

Transfer of program groups between the Chroma and Expander can be performed indirectly by saving the Chroma programs onto disk then loading them into the Expander or vice versa.

The names of program groups that were used to record sequences can be stored in the sequence file using the Comment program (User Utility Bank 1, number 0, see APPENDIX C). The names of program groups follow the syntax rules for all names outlined in APPENDIX P.

The actual structure of the program group file is detailed in APPENDIX B.

DISK CATALOG

When the system is "up" and the main menu is being displayed, type 'F' for DISK CATALOG and the catalog will appear after a second or two. The disk catalog displays the list of program groups and sequences that you have on the disk in Drive 2 if using a two drive system or Drive 1 if using a single drive system. If there are other files on the disk (as will definitely be the case in a single drive system) the disk catalog simply does not display them because they are not new TYPE A or new TYPE B files.

At the top of the catalog on the left is the Volume number. This is useful for keeping track of several diskettes. The diskette can be assigned a volume number when you initialize it for sequence storage. Read your APPLE manuals and APPENDIX M to learn how to do this. The Sequencer pays no attention to volume numbers, it just relays that information for your use. The next item at the top of the disk catalog is the number of free sectors left on the diskette. This indirectly tells you how many more sequences and program groups you can store on this disk. A 3.3 DOS diskette has available 496 sectors for sequence and program group storage (minus two sectors for a "HELLO" program). A single drive system has considerably less (approximately 300 sectors) since the Sequencer operating system must reside in Drive 1.

Each entry of the catalog also contains useful information. On the extreme left is the File Type. "P" means that this entry is a Program Group, which is a group of 50 programs much like a "fast dump" section of a cassettte for the Chroma. "S" means that this entry is a Sequence. The next item in the catalog entry is the number of sectors that this file occupies on the diskette. This information in addition to the free sector amount at the top of the catalog can be used to determine how many more sequences or program groups you can store. Program groups consisting of 50 programs always take 18 sectors. The next and last item in the entry is the file name which is used for loading,saving and deleting the files.

After the number of entries reaches the maximum allowed on the screen at any one time, the Sequencer asks you to type any key to continue. Once you get to the end of the catalog, type any key to return to the main menu.

DELETING A SEQUENCE

To delete a sequence from disk, select **D** from the main menu. The Sequencer will then ask if you want to see the disk catalog. Type **Y** if you do, any other key if you don't. After the Catalog is displayed, press any key to continue. The Sequencer will then ask for the name of the sequence to delete. You can abort and return to page one of the main menu at this time by typing <RET>. Otherwise, carefully type the name of the sequence then <RET>. At this point the Sequencer will warn you that you are about to delete data from the disk and ask for verification. Type **Y** to continue DELETE or any other key to abort. If all goes well (meaning that the file was found, etc.), the Sequencer will tell you that the sequence was deleted and will display page one of the main menu.

WARNING: Neither Apple DOS nor the Sequencer checks file types when deleting files. You can, therefore, inadvertently delete a Program Group file from the DELETE a sequence menu selection or vice versa. It is important to never name a sequence the same as a Program Group. The Sequencer will create two separate files of the same name as long as the files types are different. When deleting a file, however, the Sequencer will delete the first file it finds in the Catalog with that name, regardless of the file type. If you are operating a single drive system, it is imperative that you never use the following names for sequences or Program Groups:

HELLO	RECORD TRACK.1
CHROMA.BEGIN	PAGE 3.HEX
SEQUENCER.HEX	EDITOR.HEX
APPLE MONITOR	CLICK.HEX
TRACK.DIR.HEX	MESSAGES.HEX
SYS.PORT.HEX	FIND.SLOT.HEX
MENU3.HEX	TRANSFER.HEX

or USERxx.HEX where xx is any 2 digit number

SAVE A SEQUENCE

To save a sequence, type **S.** The Sequencer will display the name of the sequence and ask if you want to rename it. If the sequence has no name the Sequencer will disply the name as "". If you change your mind about saving the sequence, type ESC at this time and you will get back to the main menu. If you do not want to change the name of the sequence, type any key except **Y.** The sequence must be named before it can be saved.

Type **Y** if you do want to change the name before saving. The Sequencer will then ask you to type in the new name. If you type RETURN at this point, the sequence will be saved with the old name (in other words, renaming is aborted). Legal sequence names are those that follow the name syntax rules of APPENDIX P. You should read the WARNING in the DELETING A SEQUENCE section above regarding file names.

When the sequence is saved, various things are recorded within the first 512 bytes of the sequence file that reflect the system status, such as the click track setting, instrument ID's revision numbers and the timer source and increment, etc. All of this is detailed in APPENDIX B.

GET A SEQUENCE

Loading a sequence from disk will erase any existing sequence, so you may want to save the old one as described in this chapter. To get a sequence, select **G** from the main menu. The Sequencer will then ask if you want to see the disk catalog. Type **Y** if you do, any other key if don't. After the catalog is displayed, type any key to continue. The Sequencer will then ask you to type in the name of the sequence followed by a <RET>. You can abort and return to the main menu at this point by typing <RET> instead of a file name. Otherwise, type in the name of the sequence followed by a return. If all goes well (meaning that the file was found, etc.), the Sequencer will tell you that the sequence was loaded and will return to the main menu.

There will be an inverse video space after the **G - GET SEQUENCE** menu selection and the MEMORY USAGE will display the percentage of total RAM occupied.

Loading a sequence will also set the Click Track and the Timer Source and Increment that was set-up at the time the sequence was last saved.

THE CLICK TRACK

The click track is really not a track at all. It is a hardware filtered pulse from the Interface PC board, which is controlled by the Sequencer software. This chapter describes how the click track is generated and used in the Sequencer, how to set it up and what restrictions apply to changing it once set up. This chapter also explains time signatures, the MEASURE command concept and how the Sequencer determines the loop time of a sequence.

HOW IT IS GENERATED

The Sequencer uses the BPM value input by you, among other things, in determining when to toggle a flip/flop on the Interface PC board. To emphasize the first beat of the measure, the Sequencer reduces the energy of subsequent beats by toggling the flip/flop twice (approximately 28 microseconds apart).

NOTE : The click output hardware is high impedance (appr. 10K). If you do not use an amplifier or mixer input that has an input impedance of at least 100K ohms, you may hear differences between each measure. You may even not be able to hear the emphasis on the first beat of the measure. If you must use that particular amplifier input, an authorized Rhodes Service Center may be able to change the output impedance of the Interface PC board slightly to accomodate you. Call the Rhodes Chroma Service Department at (617) 938-1610 for details.

THE NEED FOR A CLICK TRACK

Besides the obvious reason of helping you to keep the beat, the click track is used to determine measures in the sequence and to calculate the loop time. Loop time is calculated for you, as described below, as long as there is a click track that has been used for at least the first track record operation.

The click track may also be used in the recording studio to provide a stable audio reference for other instruments you may want to record. Presently, the Sequencer cannot "sync" off a pre-recorded version of its click track because the frequency is way too low. See APPENDIX F for details.

TIME SIGNATURE

Time signatures in the click track can be anything from 1/1 to 15/15, although the musical value of some of these are questionable. The lower part of the signature (which type of note gets a beat) is not used by the Sequencer; but it is included for your information. The Sequencer will not syncopate any time signature, but the first beat of the measure can be emphasized.

Time signatures are required if you set up a click track. They are used to tell the Sequencer when a measure boundary has ocurred and when to loop back to the beginning of the sequence when you are in loop mode.

HOW TO SET IT UP

If a click track is to be used in a sequence, it MUST be set up prior to recording the first track. To set it up, select **K** from the main menu. After the overlay is loaded, the Sequencer will ask if you want a click track. The need for this question will become clear later, but for now, type **Y**. The Sequencer responds by asking for the time signature. Anything from 1/1 to 15/15 is allowed. The Sequencer, however, doesn't use the lower portion (which type of note gets a beat). Enter the signature as you would write it on paper, i.e. **4/4** then <RET>. Typing <RET> instead of a time signature defaults to the current signature. When the Sequencer is turned on, the click track is set up at 60 BPM and 4/4 even though the indicator on page one of the main menu says NONE.

After you select a time signature, the Sequencer will display a small menu that looks something like this:

```
CURRENT SPEED IS 60 BPM

                METRO MENU

        XXX - INPUT BPM (34-234) THEN <RET>

              P - PLAY CLICK TRACK

                <ESC> - MAIN MENU

PLEASE SELECT...
```

The first selection (XXX) allows you to just type the BPM and press <RET>. If you type **P** to play the click track, the Sequencer will start to play the click as you have set it up so far and will display the following menu:

U - SPEED UP

D - SLOW DOWN

<ESC> - METRO MENU

PLEASE SELECT...

Typing **U** will speed up the click track slightly. Use the <REPT> key at the same time as **U** and the click track will speed up continuously. The same goes for **D** to slow the click track down. When you are satisfied with the speed, type <ESC> to exit back to the METRO MENU. You will now see the same menu as before, except notice that the BPM value has been changed to the current value. Type <ESC> again to return to the main menu. Notice that the inverse video click track setting indicator after the menu selection **K - CLICK TRACK** says that the click track is ON and displays what you have set up.

NOTE : The speed of the click track is effected by the TIMER SOURCE, TIMER INCREMENT and the SPEED CHANGE pedal position (if ON). The perceived speed of the click track may not, therefore, be equal to the BPM shown. The BPM indicators are correct only if the Sequencer is driven by an INTERNAL clock (or an EXTERNAL clock of exactly 1000 Hz) with a TIME INCREMENT of 1X and the SPEED CHANGE switch OFF.

NOTE : If the SPEED UP or SLOW DOWN selections are used to set up the speed, the BPM display will round the BPM to the nearest unit. The actual speed, however, will be what you hear.

RESTRICTIONS ON CHANGING THE CLICK TRACK

When the Sequencer is powered up or the sequence is cleared, the BPM is set at 60 and the time signature is set at 4/4. Other than these default values being set up, the Sequencer acts just as if there is no click track (NONE).

When the **^X - DELETE ALL MEASURES** selection is made from the main menu, the Sequencer erases all measure commands (described below) as well as all traces of a click track. This sets up the click track as if you cleared the sequence (NONE).

Once the first track of a sequence has been recorded, the only thing you can do to a click track is turn it ON and OFF. There is a difference between OFF and NONE. NONE means that there was never a click track and the Sequencer can not store measure commands and may not loop in perfect time. This is not true if the click track is merely OFF, even if you extend the endpoints of sequences during record operations.

To turn a click track ON or OFF once a track has been recorded, just select **K** from the main menu. When the Sequencer asks if you want a click track, press **Y** or **N**.

When a sequence is loaded from disk, the click track is set exactly the way it was when the sequence was last saved.

MEASURE COMMANDS

Measure commands are stored by the Sequencer right at the first beat of the measure, starting with MEASURE 1. These commands are used as ENDPOINT references and as markers in the Editor.

MEASURE commands are stored while recording subsequent tracks only if you extend the length of the sequence. This will occur whether the click track is ON or OFF, but will never occur if the click track is NONE.

If you play the downbeat slightly before the first beat of the measure, that note will logically fall into the previous measure. When this happens and you reference the measure for an ENDPOINT, you may not hear the down beat. If this occurs on the first note of a sequence, it will also effect the loop time, as explained below. This problem can be remedied by moving the ATTACK command right after the MEASURE command via the Editor.

LOOP TIME

The Sequencer will correctly determine the loop time if the first track of a sequence was recorded with a click track and you follow a few simple rules when beginning and terminating all track record operations. If these rules are not followed and/or you reference ENDPOINTS other than the actual beginning and ending of the sequence, the Sequencer may have to add a measure's worth of time to the loop. The end result will be a full measure rest between loops. Looping with a slow EXTERNAL clock requires a little more effort as described in APPENDIX F.

When you record the first track, the Sequencer takes note of the time the first event occurs relative to the click pulse time and the beat of the measure. This information is saved with the sequence and is not updated unless you change it by recording a note on a subsequent track before the first event of the first track. By the way, the only way you can do this is by recording subsequent tracks with the click track ON, since it is only under this condition that the Sequencer starts the recording mode one measure before the start of the sequence.

When the last event is played in the sequence, the Sequencer loops back around and presets the timer to a value that will cause the first note to be played in time. The last event in this case is not a RELEASE but when you press the space bar or footswitch to terminate the record operation.

When you press the footswitch or space bar, the Sequencer will store a TIME command followed by an UNDEFINE command, which defines the end of the track. This is required because you may be using a voice with a long release time constant (the UNDEFINE command squelches the instrument).

The main rule regarding loop time is that you must allow enough time for the Sequencer to do its job. The last event in a sequence must occur such that there is sufficient time left in the measure to play the first note on time. Sometimes this requires the use of the Editor to delay the first event time or pull back the last event time.

Most of the time, it is easy to follow these simple rules while you are recording. If you need to touch it up, however, MEASURE commands stored in the sequence are a good indication of where these events must occur in time. In general, if you want to start the sequence on the down beat, the first event must occur slightly after the down beat and the last event must occur slightly before the down beat of the next measure. Give the Sequencer a few TIME INCREMENTS between each end to accomplish its task.

Again, if you do not follow these rules, the worst that can happen is that the loop time will be increased by a measure's worth of rest.

If a click track is not used, the only thing determining loop time is the first event and when you press the footswitch or space bar at the end of the record operation. There will be a slight processing delay for which you will have to learn to compensate. This delay will be constant for a given length of sequence. The Sequencer will take care of this for you only if you use a click track.

When using an EXTERNAL clock, loop time requires special considerations that are explained in APPENDIX F.

RECONFIGURE

The **X - RECONFIGURE** selection from the main menu allows you to change certain operating parameters of the Sequencer. These paramenters involve the timer, SYNC input, FOOTSWITCH input, click track, Expander port, disk drives and the Interface PC board slot. RECONFIGURE, therefore, is very useful when installing your Interface Kit and when attempting to interface the system to drum machines and other sequencers.

When the selection is made, the following menu will appear:

```
RECONFIGURE:

     C - CHANGE THIS SET-UP

     S - SAVE THIS SET-UP

     L - LOAD SET-UP

     V - VIEW CURRENT STATUS

<ESC> - MAIN MENU

     CAUTION : THESE SELECTIONS SHOULD BE
     MADE WITH A KNOWLEDGE OF THE SEQUENCER
     SYSTEM - CONSULT YOUR OWNERS MANUAL!

PLEASE SELECT...
```

This chapter will explain each of these RECONFIGURE selections in detail. When you want to exit to page one of the main menu, just type **<ESC>** when this menu appears.

VIEW CURRENT STATUS

This selection enables you to view the way the parameters are currently set-up. It also allows you to learn what the various software revisions there are in the system and the connection status of each port.

A detailed explanation of each entry in the status display follows:

CHECK SYNC? N

This line tells you if the Sequencer will wait for positive going pulses on the SYNC input before starting to play. If CHECK SYNC? is Y, then the Sequencer will display the following when you select PLAY or PLAY ALONG from the main menu:

PLAYING...WAITING FOR SYNC...

When the required number of pulses are received, the Sequencer will start playing. The WAITING FOR SYNC... message will not be cleared. The Sequencer will not wait for SYNC pulses on subsequent loops.

WAIT FOR HOW MANY? 0

This is displayed only if CHECK SYNC? is Y. The number represents the number of pulses -1 that must occur before the Sequencer will start playing. If you are using a footswitch in the SYNC input, it must be TTL compatible (must not be hardware debounced). The number of wait pulses should be 0 in this case because of switch bounce. The range allowed is 0 to 15. The timing restrictions are specified in APPENDIX O.

CHECK FOOTSWITCH? Y

The footswitch is used to tell the Sequencer when to start recording and when to stop playing or recording. The debounce software requires that the Sequencer know if there is a footswitch connected in the FOOTSWITCH input. If you plan to use the Sequencer without the footswitch connected, you must set this to **N** or the Sequencer will not PLAY a sequence.

EMPHASIS ON FIRST BEAT? Y

If this is **Y**, then the click track will emphasize the first beat of the measure.

TIMER SOURCE : INT

The timing information can currently come from 4 sources:

SYMBOL	SOURCE	ACCESS
INT	Internal	I
EXT	External	E or X
SS1	Single Step1	S
SS2	Single Step2	^S

Single Step1 currently is not used (doesn't work). Single Step2 is designed to increment the timer every low to high transition on the SYNC input. It can sense TTL pulses of greater than 3 milliseconds or a footswitch in the SYNC input. If a footswitch is used, the timer will increment when the footswitch is depressed (if the footswitch is a normally closed one).

TIME INCREMENT : 1X

The TIME INCREMENT is the amount of increment in the time value for every timer source pulse. The options are 8X, 4X, 2X, 1X, 1/2X, 1/4X, 1/8X and 1/16X. When accessing these options (in CHANGE THIS SET-UP), you need not input the X. This parameter allows a quick method of doubling or halfing the speed of the sequence while keeping the click track intact. It is also useful when converting a sequence to run on a TIMER SOURCE that is different than the one on which it was recorded.

SEQUENCER REVISION? 3

This is the software revision of the Sequencer program you are running.

THIS SEQ REVISION? 3

This is the software revision of the Sequencer program that was used the last time this sequence was saved. Essentially, it is the software revision of the sequence. The file structure of sequences are designed to be transportable. In other words, you can send a song to a friend! This sort of information is useful when there are several revisions of Sequencers in the field.

THIS SEQUENCE INSTRUMENT? Chroma/Expand

This is the instrument(s) that was used the last time this sequence was saved. The first instrument tells what was connected to the Chroma port and the second one tells what was connected to the Expander port.

THIS SEQUENCE INSTRUMENT REVISION? 2/2

This is the interface software revision of the instrument(s) that was used the last time this sequence was saved. "REV 12" Chroma software has an interface software revision of 2. "REV 13" Chroma software will have an interface software revision of 3.

CHROMA PORT STATUS? Chroma, REVISION? 2

This is the instrument and its interface software revision that is currently connected to the Chroma port.

EXPANDER PORT STATUS? Expander
REVISION? 2 AUTO REROUTE? N

This is the instrument and its interface software revision that is currently connected to the Expander port. The AUTO REROUTE parameter, when Y, allows the Sequencer to automatically send a track to the Expander port whenever you attempt to output more than 8 tracks to the Chroma port.

DISK SLOT? 6

This is the current Disk Controller Slot number. The Sequencer will support only one disk controller. It determines the slot when you power up and provides this information for your reference.

NUMBER OF DRIVES? 2

This is the number of disk drives that are connected to the system. If it is 1, then the Sequencer will direct all disk accesses to Drive 1. If it is 2, then the Sequencer will access Drive 2 for catalog, sequences and program groups and Drive 1 for RECONFIGURE set-ups.

INTERFACE SLOT? 5

This is the Interface PC board Slot number.

CHANGE THIS SET-UP

This selection allows you to change several of the parameters described in VIEW CURRENT STATUS above. When the colon appears, you may select any of the options that apply to that particular parameter. In this manual, the options are listed to the right of the colon. <DEF> means default, leave it has it is, and is selected by typing <RET>.

CHANGE SET-UP...

CHECK SYNC? : Y, N, <DEF>

WAIT FOR HOW MANY? : 0 through 15, <DEF>

CHECK FOOTSWITCH? : Y, N, <DEF>

EMPHASIS ON FIRST BEAT? : Y, N, <DEF>

TIMER SOURCE : : INT, EXT, SS1, SS2, <DEF>

TIME INCREMENT : : 8X,4X,2X,1X,1/2X,1/4X,1/8X,1/16X,<DEF>

AUTO REROUTE? : Y, N, <DEF>

NUMBER OF DRIVES? : 1, 2, <DEF>

INTERFACE SLOT? : 2 through 7

NOTE : WAIT FOR HOW MANY? is displayed only if checking SYNC.

NOTE : In TIME INCREMENT, you do not have to type the X.

SAVE THIS SET-UP

This allows you to save the changed set-ups for easy recall later. If you save it under the name of **PAGE 3.HEX**, then that set-up will be loaded upon power up. The names must follow the name syntax as outlined in APPENDIX P. You should also read the WARNING regarding names in DELETING files in Chapter 7. The Sequencer saves the set-up as a binary file on Drive 1, regardless of the number of drives in the system. Since there is no RECONFIGURE set-up catalog, you should name the set-ups such that they will be easily recognized when you catalog drive 1.

LOAD SET-UP

This allows you to load presaved set-ups immediately. There is, unfortunately, no menu selection that allows you to catalog these set-ups. You can find set-ups by selecting <ESC> to FP from the main menu, then typing CATALOG,D1. All files on Drive 1 will be displayed. Type GOTO 160 to rerun the Sequencer, then select **X** to return to RECONFIGURE. If you load PAGE 3.HEX, the power up set-up will be loaded.

The section of VIEW CURRENT STATUS that deals with revisions and port status may not reflect actual current conditions until you REINITIALIZE the Sequencer, as described in Chapter 10. You need not REINITALIZE for the Sequencer to work at this time because the system reinitializes itself at various points in the software.

DELETING SET-UPS

The Sequencer currently does not support the deletion of RECONFIGURE set-ups. If you must delete them (to get more disk space, etc.) you can do the following:

Select **<ESC>** to FP from the main menu,
Type **CATALOG,D1** then,
When you have found the set-up file, **DELETE** it,
Type **GOTO 160** to rerun the Sequencer.

Typing GOTO 160 instead of RUN will keep your sequence intact.

MISCELLANEOUS MENU SELECTIONS

This chapter describes some miscellaneous main menu selections that do not really fit into the other chapters.

H - CHROMA<>EXPANDER

This selection allows the direct transfer of data from the Chroma port to the Expander port and vice versa. The normal SET-SPLIT functions effecting interface communication on each instrument operate as normal. They are listed below for your convenience:

SET SPLIT NUMBER	FUNCTION
16	Turn OFF Performance Information
17	Turn ON Performance Information
18	Turn OFF Panel Information
19	Turn ON Panel Information
34	Turn OFF Pressure Information
35	Turn ON Pressure Information
20	Send Current Program 0

I - REINITIALIZE

This selection reinitializes the Sequencer and, to some extent, both the Chroma and Expander. It attempts to re-establish communication by outputting 260 NOP commands to each port. If the instruments are not totally confused, it will sync them to the Sequencer. It also sends ID commands to each port and finds out whether they are a Chroma or Expander and their software revisions. The Sequencer does not tell you what it found, but you can view the result from RECONFIGURE, VIEW CURRENT STATUS.

Some of the other things that REINITIALIZE does is described below:

-sets the IRQ and BRK Vectors
-clears the I/O queues
-sends, receives and discards a NOP from each port
-disables output interrupts, enables input interrupts
-clears the timer interrupt
-sets the timer according to TIMER SOURCE
-initializes the TIME INCREMENT
-Starts the ADC
-Determines whether the FOOTSWITCH and SYNC inputs are normally high or low (NC or NO for footswitches)

ERROR CONDITIONS AND CODES

There are five different types of error conditions that can occur in the Sequencer:

(1) Those that are fatal, meaning those conditions in which there is a possibility of loosing data.

(2) Those that are not fatal but may or may not stop the current operation.

(3) Errors that occur in the Editor, such as syntax errors.

(4) File Management or disk errors.

(5) Power up error conditions.

FATAL ERRORS:

1 SYNC ERR — The Sequencer was expecting an echo response from Chroma but got something else. This condition can occur in many points in the software.

10 SYNC ERR — This will occur if the Chroma does not correctly echo the PANEL SWITCH OFF command following a record operation.

12 SYNC ERR — This will occur if the Chroma does not correctly echo the STATUS1 command preceeding a record operation.

6 DEFINITION ERR — This will occur if a command encountered by the Sequencer references a Chroma instrument that has not been previously defined.

NONFATAL ERRORS:

Four conditions can occur that are not really errors but are exceptions to the normal. These are:

(1) Sequence Memory Full- you have used all available memory for this sequence. Data recorded up to that point is still intact. Recording is terminated and the Sequencer returns to the main menu.

(2) Out Of Instruments- you are attempting to output more than 8 tracks to the Chroma. The track that put you over the li mit will be muted and the Sequencer will continue to record or play. If the AUTO REROUTE is enabled this condition cannot occur because up to 8 tracks are automatically rerouted to the Expander whenever more than 8 tracks are sent to the Chroma.

(3) No Expander- You have attempted to send a track to an Expander that is not on line. The track will be muted and the Sequencer will continue to record or play.

(4) Clean Sequence- If you try and perform an operation that cannot be completed because there is no sequence in memory, the Sequencer will return to the main menu.

OTHER NONFATAL ERRORS ARE:

ERRCOD CODES: 10-call upon SEARCH without an EOS command at start of sequence.

11-illegal SEARCH OPERATION call type.

18-measure not found in monitor.

EDITOR SYNTAX ERROR CODES:

12 - unrecognized command in disassemble routine.

13 - syntax error in entering command in the editor.

14 - user attempts to change Define command in editor. This is currently illegal.

15 - user attempts to delete EOS command.

16 - track out of range or not in edit mode in editor.

DISK ERRORS:

3- write protected

4- end of data error

5- file not found

6- volume mismatch

7- I/O error

8- disk full

9- file locked

10-syntax error

11-no buffers available

12-file type mismatch

13-program too large

14-not direct command

15-unrecognized error

OTHER DISK ERRORS:

The Sequencer will print specific messages when the data diskette is full, if the overlay called upon was not found or the sequence or program group file was not found.

POWER UP ERRORS:

The Sequencer will attempt to talk to a Chroma on the Chroma port and an Expander on the Expander port. If communication on the Chroma port does not occur the Sequencer will tell the user to check the cables and try again. If communication occurs but the instrument is not a Chroma and/or the software revision of the instrument is higher than expected, it will tell you so and ask if you want to continue. If communication does not ocurr on the Expander port, the Sequencer merely tells the user that the Expander is 'not on line'. If it is on line, but the instrument is not an Expander or Chroma and/or the software revision of the instrument is higher than expected, the Sequencer will tell the user and asks if he wants to continue.

As you can see, only a Chroma should be connected to the Chroma port (at this time). This is because the Chroma can send key depression data and it is the Chroma port that is referenced when the Sequencer records a track. The Expander port can except either a Chroma or an Expander. We can not say that all future products will be compatible with this Sequencer, but the large majority of them should be compatible after little more than a diskette update.

SEQUENCE AND PROGRAM GROUP FILE MANAGEMENT AND DATA STRUCTURES

Both sequences and program groups can be stored on disk in the Sequencer. Sequences are stored as 'NEW TYPE B' files and program groups are stored as 'NEW TYPE A' files. These are not Applesoft and Binary file types; they are totally different file types and cannot be accessed by normal DOS commands. Using these file types allows the catalog routine to easily determine when to print an entry. Although any file may be stored on disk, only sequences and programs will be listed when the user asks for a catalog while under control of the Sequencer. In the Sequencer catalog, file type entries for sequences are printed as an 'S' and program groups are printed as a 'P'. The file type is displayed where it is displayed in the normal Apple Dos Catalog-preceeding the length of the file. This file type arrangement also allows the Sequencer to easily distinguish between sequence and program files when loading, thereby eliminating the possibility of loading a program group into sequence RAM. The Apple DOS does not check file types when deleting a file.

Both sequences and program groups are stored and retreived using the DOS File Manager routines. This allows fast and easy storage of the normally seqmented sequence queue. It also allows grabbing one program packet at a time from the Chroma, thereby eliminating the need for a 4259 byte buffer. This is nice because the program group file formation will not interfere with the sequence or assembly language program RAM. For a comprehensive description of the DOS file manager routines, read BENEATH APPLE DOS written by D.Wirth and P.Lechner and published by Quality Software, 6660 Reseda Blvd.,Reseda,CA 91335. To my knowledge, this information is available only in that publication.

PROGRAM GROUP FILE STRUCTURE:

The structure of the program group allows up to 50 programs. The group may include only one program if desired. The data structure for the group file is as follows:

00 number of programs in the group (01 to 32 hex)
01 File type - always 20 hex
02 ID number of Rhodes instrument from which the data was received.
03 Revision number of the software of the Rhodes instrument.
04 Group type (used for group catagory searching)
05-08 Reserved for future use.
09-xx Program entries

PROGRAM ENTRY FORMAT: (85 bytes total)

00 program number
01 program type (used for searching for catagories of programs
02-10 program name (15 bytes maximum - if less then last char is 8D for cr)
11-19 not used
1A-54 program data (59 bytes)

EXPANSION OF 59 BYTE PROGRAM DATA:

ParaNo.	Group	Name	Byte(s)	7 6 5 4 3 2 1 0
0	Panel	Link Balance	31	[- - - - N N N N]
1	Control	Patch	1	[- - - - N N N N]
2	Control	Fsw Mode	5	[- - - - - N N N]
3	Control	Kybd Alg	31	[N N N N - - - -]
4	Control	Detune	2	[N N N N N - - -]
5	Control	Output Select	2	[- - - - - N N -]
6,56	Glide	Rate	28,58	[N N N N N - - -]
7,57	Glide	Shape	14,44	[- N - - - - - -]
8,58	Sweep	Mode	4,34	[- - - - - - N N]
9,59	Sweep	Rate	4,34	[N N N N N N - -]
10,60	Sweep	Rate Mod	3,33	[- - - - N N N N]
11,61	Sweep	Wave Shape	6,36	[N N N N - - - -]
12,62	Sweep	Ampl Mod	6,36	[- - - - N N N N]
13,63	Env 1	Ampl Touch	9,39	[- - - - - N N N]
14,64	Env 1	Attack	7,37	[N N N N N - - -]
15,65	Env 1	Attack Mod	7,37	[- - - - - N N N]
16,66	Env 1	Decay	8,38	[N N N N N - - -]
17,67	Env 1	Decay Mod	8,38	[- - - - - N N N]
18,68	Env 1	Release	9,39	[N N N N N - - -]
19,69	Env 2	Delay	10,40	[N N N N N - - -]
20,70	Env 2	Ampl Touch	13,43	[- - - - - N N N]
21,71	Env 2	Attack	11,41	[N N N N N - - -]
22,72	Env 2	Attack Mod	11,41	[- - - - - N N N]
23,73	Env 2	Decay	12,42	[N N N N N - - -]
24,74	Env 2	Decay Mod	12,42	[- - - - - N N N]
25,75	Env 2	Release	13,43	[N N N N N - - -]
26,76	Pitch	Tune	14,44	[- - N N N N N N]
27,77	Pitch	Mod 1 Select	18,48	[N N N N - - - -]
28,78	Pitch	Mod 1 Depth	15,45	[- N N N N N N N]
29,79	Pitch	Mod 2 Select	18,48	[- - - - N N N N]
30,80	Pitch	Mod 2 Depth	16,46	[- N N N N N N N]
31,81	Pitch	Mod 3 Select	19,49	[N N N N - - - -]
32,82	Pitch	Mod 3 Depth	17,47	[- N N N N N N N]
33,83	Width	Wave Shape	20,50	[- - - - - - N N]
34,84	Width	Width	20,50	[N N N N N N - -]

ParaNo.	Group	Name	Byte(s)	7	6	5	4	3	2	1	0
35,85	Width	Mod Select	19,49	-	-	-	-	N	N	N	N
36,86	Width	Mod Depth	21,51	-	N	N	N	N	N	N	N
37,87	Cutoff	LP/HP	15,45	N	-	-	-	-	-	-	-
38,88	Cutoff	Resonance	10,40	-	-	-	-	-	N	N	N
39,89	Cutoff	Tune	22,52	-	-	N	N	N	N	N	N
40,90	Cutoff	Mod 1 Select	26,56	N	N	N	N	-	-	-	-
41,91	Cutoff	Mod 1 Depth	23,53	-	N	N	N	N	N	N	N
42,92	Cutoff	Mod 2 Select	26,56	-	-	-	-	N	N	N	N
43,93	Cutoff	Mod 2 Depth	24,54	-	N	N	N	N	N	N	N
44,94	Cutoff	Mod 3 Select	27,57	N	N	N	N	-	-	-	-
45,95	Cutoff	Mod 3 Depth	25,55	-	N	N	N	N	N	N	N
46,96	Volume	Mod 1 Select	27,57	-	-	-	-	N	N	-	-
47,97	Volume	Mod 1 Depth	3,33	N	N	N	N	-	-	-	-
48,98	Volume	Mod 2 Select	27,57	-	-	-	-	-	-	N	N
49,99	Volume	Mod 2 Depth	5,35	N	N	N	N	-	-	-	-
50,100	Volume	Mod 3 Select	28,58	-	-	-	-	-	N	N	N
51	Panel	Link	0	N	N	N	N	N	N	N	N
52	Panel	Edit	30	N	N	N	N	N	N	N	N
53	Panel	Keyboard Split	32	N	N	N	N	N	N	N	N
54	Panel	Main Transpose	1	N	N	-	-	-	-	-	-
55	Panel	Link Transpose	1	-	-	N	N	-	-	-	-
	Sequence Program Footswitch		29	N	N	N	N	N	N	N	N
	Free bits		2	-	-	-	-	-	-	-	N
			5	-	-	-	-	N	-	-	-
			35	-	-	-	-	N	N	N	N
			14,44	N	-	-	-	-	-	-	-
			16,46	N	-	-	-	-	-	-	-
			17,47	N	-	-	-	-	-	-	-
			21,51	N	-	-	-	-	-	-	-
			22,52	N	N	-	-	-	-	-	-
			23,53	N	-	-	-	-	-	-	-
			24,54	N	-	-	-	-	-	-	-
			25,55	N	-	-	-	-	-	-	-

Presently, the sequencer will store and retreive Groups consisting of 50 programs only. The first byte of the program name field will be 8D to signify that it is not named. Types (both group and program) will be 00 signifying sequence programs.

A Main Menu selection "Q - Program Files" will be provided. When selected, the sequencer will display the following menu:

PROGRAM FILE MANAGEMENT

1 - Save Chroma programs
2 - Get programs from disk and send to Chroma
3 - Delete program group from disk
and if there is an Expander on line:
4 - Save Expander programs
5 - Get Expander programs from disk

SEQUENCE FILE STRUCTURE:

The sequence file is structured as a block of actual sequence data precceded by a two page global variable section. The sequence data always begins and ends with an EOS command (opcode 00 followed by two bytes of time data). The absolute location of the start of the file in RAM is contained in MANSEQ (zero page locations $1E and $1F). In the controlling BASIC program this variable is called I%, so in this file description, we will call the beginning of the file I%. Following is a detailed description of the two page global variable section that preceeds the data section.

I% -INSTRUO- Last track recorded (00 signifies clean slate)

I%+1 -TYPE- File Type (always $40)

I%+2 -FIPTRL- File input pointer low (the end +1 of the data section low)

I%+3 -FIPTRH- File input pointer hi (the end +1 of the data section hi)

I%+4 -FOPTRL- File output pointer lo (the start of the data section low)

I%+5 -FOPTRH- File output pointer hi (the start of the data section hi)

I%+6 -TIMSIG- Time Signature (MS nibble is beats per measure, LS nibble is note duration, as in standard time signature notation). If METRFL =0 then TIMSIG =$44 for 4/4 time.

I%+7 -CBCRGL- Timer counts between click track outputs lo (=60000/BPM). If METRFL=0 then CBCRG=1000dfor 60 BPM.

I%+8 -CBCRGH- Timer counts between click track outputs hi.

I%+9 -METRFL- Metronome enable flag ($00= never a click track, $40= currently disabled, $C0= enabled).

I%+10 -FMEASR- Next available measure number.

I%+11 -SEQREV- Software revision of the sequencer that generated this sequence.

I%+12 -SEQNAM- Sequence name (15 bytes) - if not all 15 bytes are used, then $8D at end of name.

I%+27 -TIMINC- Timer increment (allowable values are $80,$40,$20,$10,$08,$04,$02,$01 which correspond to time increments of 8,4,2,1,1/2,1/4,1/8 and 1/16).

I%+28 -INSTBL- Instrument Table (Bit 7 mute active high, Bits 6-0 program number) 16 bytes total.

I%+44 -IVOLTB- Initial Volume Table (these are the volume values that are imbedded in the define commands) 16 bytes total. Currently not used.

I%+60 -LVOLTB- Last Volume Table (these are the last volume values of each track- to be used by continuous volume) 16 bytes total. Currently not used.

I%+76 -FTCLKL- Click time at first event in clean slate record low (used to sync click track on subsequent record operations).

I%+77 -FTCLKH- Click time at first event hi.

I%+78 -FTBEAT- The BPMEAS at first event in clean slate record (used to sync click track on subsequent record operations).

I%+79 -IDNUM- ID number of instrument on Chroma port.

I%+80 -IDNUMX- ID number of instrument on Expander port.

I%+81 -REVNUM- Software revision number of instrument on Chroma port.

I%+82 -REVXP- Software revision number of instrument on Expander port.

I%+83 -TRANTB- Transpose table (used to store transpose status of each track) 16 bytes total. Currently not used.

I%+99 -TIMSRC- Timer Source, $FF=External, $00=Internal
1KHz, $7F=Debounced Single Step,
$3F=Single Step.

I%+100 to 139- 41 bytes reserved for future use.

I%+140 -NAMTBL- Track name table, each of 16 entries
contains 15 bytes (if track name does
not use full 15 bytes, $8D is stored at
end of name).

I%+380 to 395- 16 bytes unused, reserved for future use.

I%+396 to 511- 116 byte field for comments, stored and read
through USER1, number 0 overlay.

I%+512 to FIPTR-1 Sequence data

SEQUENCE DATA FORMAT:

```
EOS $00          Marks beginning and ending of sequence data.
TIME $FF xxyy  where xx is low byte, yy is high byte.
MEASURE xx       where xx is between 01 and FF.
DEFINE $C0 tt aa bb cc dd ee ff
          where tt = Track # ($01-10)
                aa = Lever1 value ($00-FF, signed 2's complement)
                bb = Lever2 value ( same as above )
                cc = Pedal1 value ( same as above )
                dd = Pedal2 value ( same as above )
                ee = Initial Volume Value (00-FF)
                ff = Footswitches (MSB=Footsw1, next bit=Footsw2
                                   0=up, 1=down)
```

All others are stored as detailed in APPENDIX I, except that the opcode will always reflect instrument 0 and will always be followed by a track byte ($01-10).

$F8-FC and $FE opcodes are reserved for future utility commands.

PROGRAM OVERLAY STRUCTURE and USER UTILITY PROGRAMS

Do to the scarcity of RAM in the Apple system, a small section of address space in the RAM card has been designated as an overlay area. This area will contain different programs, depending on what is needed at any given time. USER UTILITY programs are provided for, should the user want to write is own routines.

Each of these programs or any routine within these programs is entered by calling a small routine (calling routine) that resides outside of the overlay area. This routine will check to see if the desired program is in memory by looking at 'OVERLAY', which will have one of the following possible values:

00 - No program in memory
01 - EDITOR in memory
02 - SET.UP.CLICK in memory
03 - MENU.PAGE.3 in memory
04 - MARKER in memory (Currently not written)
05 - RECONFIGURE in memory
06 - CVOLUME in memory (Currently not written)
07 - One program from USER UTILITY BANK 1 in memory
08 - One program from USER UTILITY BANK 2 in memory
09 - TRACK.DIR/SEARCH in memory
0A - CATALOG in memory
0B - TRANSFER in memory
0C - Reserved
0D - Reserved
0E - Reserved
0F - Reserved

If the desired program is not already in memory, the calling routine will insert the desired program code into OVERLAY and call PROGRAM.OVERLAY which will attempt to load the program. PROGRAM.OVERLAY is a file management program that reads binary files using the file manager routines set up for sequence and program group file management.

If PROGRAM.OVERLAY cannot load the desired program, it will return to the calling routine with OVERLAY set to zero, the Z flag set, and the DOS error code in the X register. At this point, the calling routine will display the error code and the message:

"SORRY, SELECTION IS NOT AVAILABLE"
"PLEASE SEE YOUR OWNERS MANUAL"

If the load is successful, PROGRAM.OVERLAY will reset the Z flag and return. At this point the calling routine will call the desired routine within the recently loaded program.

USER UTILITY PROGRAMS

There are two "banks" of user or utility programs selectable from the main menu. Each bank can include up to 10 individual overlay programs, accessed by USER NUMBERS 0-9 entered after the USER1 or USER2 selection from the main menu. The object files have the titles USER10.HEX to USER19.HEX for the USER1 bank and USER20.HEX to USER29.HEX for the USER2 bank.

USER UTILITY program overlays are always loaded regardless of whats in memory because the OVERLAY register just indicates whether a BANK 1 or BANK 2 program is in memory. The USERNO register is set to the USER NUMBER in BASIC and the overlay calling routine uses this information to load the appropriate USER overlay.

It is also possible to call a user overlay from another user overlay and return back to the original. Neither the source nor the destination addresses have to be at the beginning of the overlay.

In the case of USER UTILITY 1 and 2 overlays, the overlay calling routine will JUMP to the start of the overlay file - (OVERLAY.BASE), so the programmer should put a JMP to the start of his routine at the beginning of his object file. Currently the overlay area is approximately 2K bytes long. Care must be taken to insure that the overlay program does not write over the RESET and IRQ vectors at FFFC-FFFF.

Actual programming of these USER overlays is not difficult, since many of the variables and routines likely to be required are already set-up. Programmers must include the source file 'OVERLAY.EQUATES.SRC' in their programs to access these variables and routines. When the PROGRAMMERS MANUAL option is purchased a disk that contains this source file will be included. Also on the disk will be the same Equates in a TEXT file so that different assemblers may be used to generate the USER program. For now, however, only a listing of this file is provided in this APPENDIX. It is included mainly for the purpose of detailing Sequencer memory usage. A detailed description of each variable and routine is also included in the PROGRAMMER'S MANUAL.

Currently, there are 4 user program overlays written as examples, two in BANK 1 and two in BANK 2. They are described below:

USER UTILITY BANK 1: number0 (USER10.HEX) - COMMENT - This program allows the user to read or store a 115 character comment in the sequence. This is useful for indicating the name of the program group that was used to record the sequence and as a quick reference to the set-up.

number1 (USER11.HEX) - ECHO - This program sends multiple NOP commands to the Chroma, keeping it so busy that it will sound like tremelo or echo. It will only work on REV 2 Chromas.

USER UTILITY BANK 2: number0 (USER20.HEX) - This is a test program for the interface described in APPENDIX L.

number1 (USER21.HEX) - This is a short ADC test program to allow you to fine tune the pedal range. You can also do this in the test program described above (USER20).

SUGGESTIONS FOR USER UTILITY PROGRAMS:

- A USER UTILITY Catalog.
- A quantizing or error correcting program.
- A Program that gives a Catalog of RECONFIGURE Set-ups and allows deletion of these set-ups from disk.
- A Sequencer Data Diskette initialization/copy program.
- A track merge/replace program.

2-6 Parameters USER26.HEX
2-7 Instruments USER27.HEX

OVERLAY EQUATES PAGE 0001

```
         1010
         1020
         1030
         1040 **********************
         1050 *                    *
         1060 *  OVERLAY EQUATES   *
         1070 *                    *
         1080 **********************
         1090
         1100 *  Stored under 'OVERLAY.EQUATES.SRC'
         1110
         1120 *  VERSION 8-30-82
         1130
         1140 *  (C) COPYRIGHT 1982 CBS INC.
         1150
         1160
F69C-    1170    OVERLAY BASE EQ $F69C  Insert here
         1180
         1190
         1200 * -------------------- *
         1210 * ZERO PAGE LOCATIONS *
         1220 * --------------------
         1230
0006-    1240   DATA    .EQ $06 I/O Driver Input/Output Byte 6
0007-    1250   IWRMSK  .EQ $07 Internal Interrupt Mask Image 7
0008-    1260   CTDNOFL .EQ $08 Countdown Offset LO 8
0009-    1270   CTDNOFH .EQ $09 Countdown Offset HI 9
0018-    1280   SIPTRL  .EQ $18 SEQ-Q Input Pointer LO 24
0019-    1290   SIPTRH  .EQ $19 SEQ-Q Input Pointer HI 25
001A-    1300   SOPTRL  .EQ $1A SEQ-Q Output Pointer LO 26
001B-    1310   SOPTRH  .EQ $1B SEQ-Q Output Pointer HI 27
001C-    1340   MANSEQL .EQ $1C Start Addr of Seq File LO 28
001D-    1350   MANSEQH .EQ $1D Start Addr of Seq File HI 29
         1360
0022-    1370   WNDTOP  .EQ $22 Top Line of Scroll Window
0024-    1380   CH      .EQ $24 Horizontal Cursor Position
0025-    1390   CV      .EQ $25 Vertical Cursor Position
002A-    1400   BAS2L   .EQ $2A Scroll Operation Work Area LO
002B-    1410   BAS2H   .EQ $2B Scroll Operation Work Area HI
         1420
003A-    1430   PCL     .EQ $3A    !
003B-    1440   PCH     .EQ $3B    !
003C-    1450   A1L     .EQ $3C    !
003D-    1460   A1H     .EQ $3D    !  GP Monitor Registers
003E-    1470   A2L     .EQ $3E    !
003F-    1480   A2H     .EQ $3F    !
0040-    1490   A3L     .EQ $40    !
0041-    1500   A3H     .EQ $41    !
```

OVERLAY EQUATES PAGE 0002

```
        1510
0045-   1520   ACC1     .EQ $45 BASIC ACC Save 69
0046-   1530   BXSAV    .EQ $46 BASIC X Save 70
0047-   1540   BYSAV    .EQ $47 BASIC Y Save 71
0048-   1550   STATUS   .EQ $48 STATUS Reg Save 72
00B8-   1560   TXTPTR   .EQ $B8 BASIC Text Pointer
0049-   1570   SPNT     .EQ $49 Saved Stack Pointer
0033-   1580   PROMPT   .EQ $50 PROMPT Char ASCII
009E-   1590   COUNT    .EQ $9E Hex-Dec Count Reg Used In File Manage
0048-   1600   PREG     .EQ $48 USER STATUS Register
0032-   1610   INVFLG   .EQ $32 Video Format Control Reg
0050-   1620   LINNUM   .EQ $50 BASIC Line Number Reg
00D8-   1630   ONERR    .EQ $D8 APPLESOFT/DOS ONERR Flag
00DE-   1640   ERRNUM   .EQ $DE Error Code
000D-   1650   CHARAC   .EQ $0D BASIC String Char Reg
005E-   1660   INDEX    .EQ $5E BASIC GP Reg
        1670
        1680
        1690
        1700
        1710 **********************
        1720 *                    *
        1730 *  PAGE 3 LOCATIONS  *
        1740 *                    *
        1750 **********************
        1760
        1770
        1780
0300-   1790   STATUS0 .EQ $0300 Status and GP String Buffer
0306-   1800   STATUS1 .EQ $0306 Status and GP String Buffer
030F-   1810   SRCTYP  .EQ $030F Type of Search Operation Reg 783
0310-   1820   MNLOCL  .EQ $0310 Pointer to Monitor Routine 784
0312-   1830   SRCOP   .EQ $0312 Opcode to Search For 786
0313-   1840   SRCTRK  .EQ $0313 Track or Meas # to Search For 787
0314-   1850   SUBLOL  .EQ $0314 Loc of Subroutine After Search 788-789
0316-   1860   SYNCCK  .EQ $0316 SYNC Reg-Bit 7 Check, Bit 5 Image BITS 3-0 Inc 790
0317-   1870   FOOTCK  .EQ $0317 FOOTSW Reg-Bit 7 Check, Bit 5 Image,791
0318-   1880   CLIKEM  .EQ $0318 Click Emphasis Reg (00=EMPH, FF=NOT) 792
0319-   1890   SOURCE  .EQ $0319 GET Data Source Reg (SEQ-Q=00) 793
031A-   1900   ENDMES  .EQ $031A Ending Measure Reg 794
031B-   1910   OVERLAY .EQ $031B Indicates Current Program In Memory 795
031C-   1920   ASAVI   .EQ $031C ACC Save Reg Used by MON0 and ROMSUB 796
031D-   1930   ERRCOD  .EQ $031D Error Code for BASIC 797
031E-   1940   TIMING  .EQ $031E Timer Increment Value Reg 798
031F-   1950   INSTRB  .EQ $031F Temporary INSTRU0 Register 799
0320-   1960   VARTIM  .EQ $0320 Var Time (FF=LRG VAR, 80=SM VAR,00=NO VAR) 800
0321-   1970   STRTIM  .EQ $0321 Store Variable Time Flag (FF=STORE) 801
0322-   1980   TIMSRC  .EQ $0322 Timer Source Reg (00=INT, FF=EXT) 802
0323-   1990   SEQREV  .EQ $0323 REV Number of This Program 803
0324-   2000   IDNUM   .EQ $0324 CHROMA Port Instrument ID Number 804
0325-   2010   IDNUMX  .EQ $0325 EXPANDER Port Instrument ID Number 805
0326-   2020   REVNUM  .EQ $0326 CHROMA Port Software Revision Number 806
0327-   2030   REVXP   .EQ $0327 EXPANDER Port Software Revision No. 807
```

OVERLAY EQUATES PAGE 0003

```
0363-    2040   PRESSURE.SWITCH .EQ $0363 Pressure Record Flag (FF=ON) 867
0364-    2050   PRESS.COM.FLAG  .EQ $0364 Pressure Command Flag (FF=PRESSURE) 868
0365-    2060   CLEAN    .EQ $0365 Clean Slate Flag (CLEAN=00) 869
0366-    2070   ACCSAV   .EQ $0366 ACC Save Used by IOSAVE/REST 870
0367-    2080   RAMSTATUS .EQ $0367 RAM Card Status 871
0368-    2090   USERNO   .EQ $0368 Number of USER Overlay to Call 872
0369-    2100   REROUT   .EQ $0369 Reroute to Expander Flag (ENABLED=01) 873
036A-    2110   MEASUR   .EQ $036A Next Available Measure 874
036B-    2120   DSLOT    .EQ $036B Disk Controller Slot No. 875
036C-    2130   DDRIVE   .EQ $036C Disk Drive No. For SEQ & PROG 876
036D-    2140   CALNUM   .EQ $036D Number of Assy Call From BASIC 877
036E-    2150   LOG0     .EQ $036E MAIN Track # 878
036F-    2160   LOG1     .EQ $036F LINK Track # 879
0370-    2170   IFSLOT   .EQ $0370 Interface SLOT Number *16 880
0371-    2180   DELMES   .EQ $0371 DELAY Measure No. (NO DELAY=00) 881
0372-    2190   ASLOCL   .EQ $0372 Pointer to Assy Routine 882-883
0374-    2200   INSTRU0  .EQ $0374 Track Number Reg 884
0375-    2210   MSKTBL   .EQ $0375 Active CHROMAs Reg (00=CHROMA ONLY) 885
0376-    2220   RECPLA   .EQ $0376 Record/Play Flag (PLAY=00) 886
0377-    2230   LOOPSW   .EQ $0377 Loop ON/OFF Switch (ON=00) 887
0378-    2240   PLALNG   .EQ $0378 Play Along Flag (PLAY=00) 888
03F5-    2250   MAXLNL   .EQ $03F5 Max Ending Addr of SEQ File LO
03F6-    2255   MAXLNH   .EQ $03F6 Max Ending Addr of SEQ File HI
         2257
         2258
         2260   **** PAGE THREE ROUTINES ***
         2270
0328-    2280   BASCAL   .EQ $0328 Called by BASIC To Access Assy Routines
0346-    2290   ROMROM   .EQ $0346 Set ROM Read/RAM Write Protect
0354-    2300   RAMRAM   .EQ $0354 Set RAM Bank 2 Read/Write
0379-    2310   IRQPR    .EQ $0379 IRQ Processing Routine
         2320
         2330
         2340
         2350   * ------------ *
         2360   * I/O LOCATIONS *
         2370   * ------------ *
         2380
         2390
C000-    2400   KBD      .EQ $C000  Keyboard Character and Strobe -16384
C010-    2410   KBDSTB   .EQ $C010  Keyboard Strobe Acknowledge -16368
C020-    2420   CASS     .EQ $C020  Cassette Output Toggle Read -16352
C030-    2430   SPKR     .EQ $C030  Speaker Toggle Read -16336
         2440
         2450
0080-    2460   DISEXTC .EQ $80    WR: Disable External Clock
0080-    2470   ENEXTC  .EQ $80    RD: Enable External Clock
0082-    2480   STRADC  .EQ $82    WR: Start ADC
0082-    2490   RDADC   .EQ $82    RD: Read ADC Value
0083-    2500   CLRTIM  .EQ $83    WR: Clear Timer Interrupt Source
0083-    2510   RDMISC  .EQ $83    RD: (7) TIME -LO Indicates Time Int
         2520   *                      (6) XXFULL -LO Data Still on Port
```

OVERLAY EQUATES PAGE 0004

```
         2530   *                        (5) FOOTSW -N.O. =HI
         2540   *                        (4) Conversion Ready =LO
         2550
0084-    2560   CLICK    .EQ $84     WR: Output a HP Filtered Click
0085-    2570   XWRMSK   .EQ $85     WR: (7) XIMASK NOT
         2580   *                        (6) XOMASK NOT
         2590
0085-    2600   RDFLAG   .EQ $85     RD: (7) XIFULL NOT
         2610   *                        (6) XOFULL
         2620   *                        (5) SYNC IN - NO =HI
         2630   *                        (4) IXFULL NOT (EXPANDER Input Available=00)
         2640
0086-    2650   WREXTO   .EQ $86     WR: CHROMA Output Port
0086-    2660   RDEXTX   .EQ $86     RD: EXPANDER Input Port
0087-    2670   WREXTX   .EQ $87     WR: EXPANDER Output Port
0087-    2680   RDEXTI   .EQ $87     RD: EXPANDER Input Port
         2690
         2700
         2710   *** MACRO USED TO ADDRESS I/O ***
         2720
         2730            .MA LDIO    LOAD I/O
         2740            .DA #$00    BRK OPCODE
         2750            .DA #[2
         2760            .DA #[1
         2770            .EM
         2780
         2790   *** FORM: >LDIO (OPCODE),(I/O LABEL)
         2800
         2810   *** EXAMPLE: >LDIO LDA,RDADC
         2820
         2830   *** This loads the ACC with ADC value
         2840
         2850   * Opcodes Allowed:  ALL Are Absolute *
         2860
00AD-    2870   LDA      .EQ $AD
00AE-    2880   LDX      .EQ $AE
00AC-    2890   LDY      .EQ $AC
008D-    2900   STA      .EQ $8D
008E-    2910   STX      .EQ $8E
008C-    2920   STY      .EQ $8C
000D-    2930   ORA      .EQ $0D
004D-    2940   EOR      .EQ $4D
002D-    2950   AND      .EQ $2D
002C-    2960   BIT      .EQ $2C
00CD-    2970   CMP      .EQ $CD
00EC-    2980   CPX      .EQ $EC
00CC-    2990   CPY      .EQ $CC
006D-    3000   ADC      .EQ $6D
00ED-    3010   SBC      .EQ $ED
         3020
         3030
```

OVERLAY EQUATES PAGE 0005

```
         3040   ** MACRO Used to Save SOPTR and Other 2 Byte Pointers **
         3050
         3060          .MA SAVE
         3070           LDA [1
         3080           PHA
         3090           LDA [1 + 1
         3100           PHA
         3110          .EM
         3120
         3130          .MA RESTORE
         3140          .PLA
         3150           STA [1 + 1
         3160           PLA
         3170           STA [1
         3180          .EM
         3190
         3200
         3210
         3220   * ---------------------- *
         3230   * NORMAL BUFFER LOCATIONS *
         3240   * ---------------------- *
         3250
         3260
         3270   * HIMEM=$27EB 10219
         3280
         3290   * 27EC-95FF Sequence File Buffer (21K) 10,220
         3300
D000-    3310   OPCODE  .EQ $D000 OPCODE TABLE 2.HEX
D100-    3320   IQ.PAGE .EQ $D100 I/O Input Queue
D200-    3330   OQ.PAGE .EQ $D200 I/O Output Queue
94F5-    3340   BUFFER  .EQ $94F5 256 GP Buffer
94F5-    3360   VOLUME.TABLE .EQ $94F5
         3370
0200-    3380   LINBUF  .EQ $0200 APPLE Input Line Buffer 3390
94E8-    3400   FILE MANAGE .EQ $94E8 My Routine For Calling the FILE.MANAGER 3410
         3420
         3425
         3427
         3430   *RAM CARD VARIABLES AND SUBROUTINES IN MAIN PROG BASE*
         3440
D300-    3450   MAIN    .EQ $D300
         3460
D300-    3470   HOME    .EQ MAIN
D303-    3480   PRINT SPACES .EQ MAIN+3
D306-    3490   CROUT   .EQ MAIN+6
D309-    3500   CROUT2  .EQ MAIN+9
D30C-    3510   PRESS   .EQ MAIN+12
D30F-    3520   PRESS0  .EQ MAIN+15
D312-    3530   COUT    .EQ MAIN+18
D315-    3540   PRINT1  .EQ MAIN+21
D318-    3550   PRINT2  .EQ MAIN+24
D31B-    3560   PRINT3  .EQ MAIN+27
```

OVERLAY EQUATES PAGE 0006

```
D31E-    3570   PRINT4   .EQ MAIN+30
D321-    3580   PRINT5   .EQ MAIN+33
D324-    3590   PRINT6   .EQ MAIN+36
D327-    3600   PRINT7   .EQ MAIN+39
D32A-    3610   PRINT8   .EQ MAIN+42
D32D-    3620   PRINT9   .EQ MAIN+45
D330-    3630   PRINT10  .EQ MAIN+48
D333-    3640   PRINT11  .EQ MAIN+51
D336-    3650   PRINT12  .EQ MAIN+54
D339-    3660   GETIOB   .EQ MAIN+57
D33C-    3670   RWTS     .EQ MAIN+60
D33F-    3680   DECMAL   .EQ MAIN+63
D342-    3690   DECMAL0  .EQ MAIN+66
D345-    3700   SPACE    .EQ MAIN+69
D348-    3710   CLREOP   .EQ MAIN+72
D34B-    3720   TABV     .EQ MAIN+75
D34E-    3730   LINGET   .EQ MAIN+78
D351-    3740   GETLN    .EQ MAIN+81
D354-    3750   GETLNZ   .EQ MAIN+84
D357-    3760   PRINT.STATUS.BUFFER .EQ MAIN+87
D35A-    3770   IOSAVE   .EQ MAIN+90
D35D-    3780   IOREST   .EQ MAIN+93
D360-    3790   FPTOSP   .EQ MAIN+96
D363-    3800   SPTOFP   .EQ MAIN+99
D366-    3810   INCMNT   .EQ MAIN+102
D369-    3820   GET1     .EQ MAIN+105
D36C-    3830   GET2     .EQ MAIN+108
D36F-    3840   DELCM0   .EQ MAIN+111
D372-    3850   DELCOM   .EQ MAIN+114
D375-    3860   DELET2   .EQ MAIN+117
D378-    3870   FRTLST   .EQ MAIN+120
D37B-    3880   GET.NAME .EQ MAIN+123
D37E-    3890   WRITE.SET.UP  .EQ MAIN+126
D381-    3900   FILE.ERROR  .EQ MAIN+129
D384-    3910   IOINIT  .EQ MAIN+132
D387-    3920   CLOSE.ALL.FILES  .EQ MAIN+135
D38A-    3930   READ.SET.UP  .EQ MAIN+138
D38D-    3940   BUFFER.TO.ADDRL  .EQ MAIN+141
D390-    3950   OPEN.AND.POSITION .EQ MAIN+147
D393-    3960   METRO.SET.UP  .EQ MAIN+147
D396-    3970   METRO    .EQ MAIN+150
D399-    3980   RESOUT   .EQ MAIN+153
D39C-    3990   LINPRT   .EQ MAIN+156
D39F-    4000   CNVRT1   .EQ MAIN+159
D3A2-    4010   STRBUF   .EQ MAIN+162
D3A5-    4020   BELL     .EQ MAIN+165
D3A8-    4030   RESTORE  .EQ MAIN+168
D3AB-    4040   STAKCK   .EQ MAIN+171
D3AE-    4050   CLRACT   .EQ MAIN+174
D3B1-    4060   OUT1     .EQ MAIN+177
D3B4-    4070   EXPOUTT  .EQ MAIN+180
D3B7-    4080   EXPOUT   .EQ MAIN+183
```

OVERLAY EQUATES PAGE 0007

```
D3BA-    4090   REINIT.CHROMA.PORT  .EQ MAIN+186
D3BD-    4100   REINIT.EXPANDER.PORT  .EQ MAIN+189
D3C0-    4110   BUFOUT  .EQ MAIN+192
D3C3-    4120   PRINT0  .EQ MAIN+195
D3C6-    4130   GET.FOOTSWITCH  .EQ MAIN+198
D3C9-    4140   GET.SYNC .EQ MAIN+201
D3CC-    4150   GET.ADC.VAL  .EQ MAIN+204
D3CF-    4160   CALL.OVERLAY.FROM.OVERLAY .EQ MAIN+207
         4170
         4180
D3D2-    4190   LINE     .EQ MAIN+210
D3D3-    4200   ERRPRT   .EQ MAIN+211
D3D4-    4210   ACTDRV   .EQ MAIN+212
D3D5-    4220   YSAV1    .EQ MAIN+213
D3D6-    4230   XSAV1    .EQ MAIN+214
D3D7-    4240   TRANSP   .EQ MAIN+215
D3D8-    4250   VOLUME   .EQ MAIN+216
D3D9-    4260   PHRASE   .EQ MAIN+217
D3DA-    4270   ASMSTR   .EQ MAIN+218
D3DB-    4280   SIGN     .EQ MAIN+219
D3DC-    4290   MON1     .EQ MAIN+220
D3DD-    4300   STMEAS   .EQ MAIN+221
D3DE-    4310   BPMEAS   .EQ MAIN+222
         4320
D3DF-    4330   XRDFRC   .EQ MAIN+223
D3E0-    4340   XRDLST   .EQ MAIN+224
D3E1-    4350   XRDMST   .EQ MAIN+225
         4360
D3E2-    4370   FIXTMFRC .EQ MAIN+226
D3E3-    4380   FIXTML   .EQ MAIN+227
D3E4-    4390   FIXTMH   .EQ MAIN+228
         4400
D3E5-    4410   TTIMEL   .EQ MAIN+229
D3E7-    4420   LASTML   .EQ MAIN+231
D3E9-    4430   TEXTPL   .EQ MAIN+233
D3EB-    4440   CLIKTML  .EQ MAIN+235
D3ED-    4450   TEMPOL   .EQ MAIN+237
D3EF-    4460   ADDRL    .EQ MAIN+239
         4470
D3F1-    4480   BUFPNT   .EQ MAIN+241
D3F2-    4490   COMBUF   .EQ MAIN+242
D3FA-    4500   PORT     .EQ MAIN+250
         4510
D3FB-    4520   CLIK.OFF  .EQ MAIN+251
D3FC-    4530   COUNTL   .EQ MAIN+252
D3FD-    4540   COUNTH   .EQ MAIN+253
D3FE-    4550   DEBUG    .EQ MAIN+254
D3FF-    4560   MUTFLG   .EQ MAIN+255
D400-    4570   LINEL    .EQ MAIN+256
D402-    4580   PLAEND   .EQ MAIN+258
D403-    4590   INSTRU   .EQ MAIN+259
D404-    4600   RECSW    .EQ MAIN+260
```

OVERLAY EQUATES PAGE 0008

```
D405-    4610   ENDTML  .EQ MAIN+261
D407-    4620   UNDEFTM .EQ MAIN+263
D409-    4630   MEASTM  .EQ MAIN+265
D40B-    4640   BITEST  .EQ MAIN+267
D40C-    4650   XIIN    .EQ MAIN+268
D40D-    4660   XIOUT   .EQ MAIN+269
D40E-    4670   X01N    .EQ MAIN+270
D40F-    4680   XOOUT   .EQ MAIN+271
D410-    4690   CTIMEL  .EQ MAIN+272
D411-    4700   CTIMEH  .EQ MAIN+273
D412-    4710   ITIMINC .EQ MAIN+274
D414-    4720   TRYAGN  .EQ MAIN+276
D415-    4730   SYNCIM  .EQ MAIN+277
D416-    4740   SYNC.COUNTER  .EQ MAIN+278
D417-    4750   FOOTIM  .EQ MAIN+279
D418-    4760   BOUNCE  .EQ MAIN+280
D419-    4770   IFIXINC .EQ MAIN+281
D41B-    4780   LOCAT   .EQ MAIN+283
D41C-    4790   LAST.ADC.VAL  .EQ MAIN+284
         4880
         4810
D41D-    4820   MEMFUL0 .EQ MAIN+285
D420-    4830   FIND.LENGTH  .EQ MAIN+288
D423-    4840   CLEAR.SEQUENCE.ASSY  .EQ MAIN+291
D426-    4850   INIT.FILE.PNTRS  .EQ MAIN+294
D429-    4860   PRINT.SEQ.NAME  .EQ MAIN+297
D42C-    4870   PRINT.NAME  .EQ MAIN+300
D42F-    4880   GET.Y.N.  .EQ MAIN+303
D432-    4890   GET.PROG.DATA  .EQ MAIN+306
D435-    4900   SECOND.CHANCE  .EQ MAIN+309
D438-    4910   CATALOG.AND.ASK.FOR.NAME  .EQ MAIN+312
D43B-    4920   XFER.LINBUF.TO.STATUS .EQ MAIN+315
D43E-    4930   XFER.SEQNAME.TO.STATUS  .EQ MAIN+318
         4940
D444-    4950   FIND.FREE.BUFFER  .EQ MAIN+324
D447-    4960   SET.UP.OPEN  .EQ MAIN+327
D44A-    4970   POSITION.AT.START  .EQ MAIN+330
D44D-    4980   SEND.PROG.TO.PORT  .EQ MAIN+333
D450-    4990   READ.PROG.GROUP  .EQ MAIN+336
D453-    5000   SAVE.PROGRAMS  .EQ MAIN+339
D456-    5010   INSEQ0  .EQ MAIN+342
D459-    5020   INIT    .EQ MAIN+345
D45C-    5030   INSEQ   .EQ MAIN+348
D462-    5050   INSEQ5  .EQ MAIN+354
D465-    5060   INSEQ4  .EQ MAIN+357
D468-    5070   INSEQ2  .EQ MAIN+360
D46B-    5080   INSEQ6  .EQ MAIN+363
D46E-    5090   STORE1  .EQ MAIN+366
D471-    5100   STORE   .EQ MAIN+369
D474-    5110   STORE0  .EQ MAIN+372
D477-    5120   INDEC   .EQ MAIN+375
D47A-    5130   OUTDEC  .EQ MAIN+378
```

OVERLAY EQUATES PAGE 0009

```
D47D-    5140   SERCHI   .EQ MAIN+381
D480-    5150   GETPR0   .EQ MAIN+384
D483-    5160   OUTPUT   .EQ MAIN+387
D486-    5170   NEXCOM   .EQ MAIN+390
D489-    5180   NOMEAS   .EQ MAIN+393
D48C-    5190   MORDAT   .EQ MAIN+396
D48F-    5200   GET.TIME.VALUE   .EQ MAIN+399
D492-    5210   QSTAK0   .EQ MAIN+402
D495-    5220   STRSTA   .EQ MAIN+405
D498-    5230   CHRGET   .EQ MAIN+408
D49B-    5240   INPUT1   .EQ MAIN+411
D49E-    5250   INPUT    .EQ MAIN+414
D4A1-    5260   INPUT1A  .EQ MAIN+417
D4A4-    5270   COMP1    .EQ MAIN+420
D4A7-    5280   COMP.PORT   .EQ MAIN+423
D4AA-    5290   EXPOUTD  .EQ MAIN+426
D4AD-    5300   INPUTXA  .EQ MAIN+429
D4B0-    5310   INPUTXW  .EQ MAIN+432
D4B3-    5320   INPUTBB  .EQ MAIN+435
D4B6-    5330   IN.PORT.B   .EQ MAIN+438
D4B9-    5340   IN.PORT.A   .EQ MAIN+441
D4BC-    5350   OUT.PORT.A  .EQ MAIN+444
D4BF-    5360   REINIT   .EQ MAIN+447
D4C2-    5370   SWITCH   .EQ MAIN+450
D4C5-    5380   GTLINK   .EQ MAIN+453
D4C8-    5390   CALCULATE.MEASTM   .EQ MAIN+456
D4CB-    5400   CALCULATE.FTBEAT.M1   .EQ MAIN+459
D4CE-    5410   CALCULATE   .EQ MAIN+462
D4D1-    5420   PROGRAM.OVERLAY   .EQ MAIN+465
D4D4-    5430   GET.OVERLAY   .EQ MAIN+468
         5440
D4D7-    5450   DELAY    .EQ MAIN+471
D4D8-    5460   SEPASS   .EQ MAIN+472
D4D9-    5470   KYCHAR   .EQ MAIN+473
D4DA-    5480   USER1.SPACE   .EQ MAIN+474
D4DC-    5490   USER.SPACE   .EQ MAIN+476
         5500
         5510
D4DD-    5520   PRBYTE   .EQ MAIN+477
D4E0-    5530   GET.SYNC0   .EQ MAIN+480
         5540
         5550
D4E3-    5560   ACTBL0   .EQ MAIN+483
D4EB-    5570   ACTBL1   .EQ MAIN+491
         5580
D4F3-    5590   CLREOL   .EQ MAIN+499
         5600
D4F6-    5610   TIMCOMP  .EQ MAIN+502
```

OVERLAY EQUATES PAGE 0010

```
D4F7-    5620   SSIMAGE .EQ MAIN+503
D4F8-    5630   SSIMAGE0 .EQ MAIN+504
         5640
D4F9-    5650   TRANSP0 .EQ MAIN+505
D4FA-    5660   TRANSP1 .EQ MAIN+506
D4FB-    5670   TRANSP2 .EQ MAIN+507
D4FC-    5680   TRANSP3 .EQ MAIN+508
         5690
D4FD-    5700   TRACK.TABLE .EQ MAIN+509 GP Track Table (Used by EDITOR)
         5710
         5720
```

SYSTEM MEMORY USAGE

The following is a description of the way the Sequencer uses the Apple memory address space. The Apple itself uses a lot of the space for things such as the Monitor and BASIC Interpreter ROM, stack, input line buffer, DOS image, screen text, etc. The Sequencer uses many of the Apple functions and, therefore, uses this space as well. This section, however, will concentrate on memory used exclusively for the Sequencer.

OVERVIEW:
The controlling program of the Sequencer (RECORD TRACK.1) is an Applesoft BASIC program that resides from $0800 to $27EB. This program calls assembly language routines that reside in the 16K bank switched RAM card required for the Sequencer System. The actual sequence data resides in non-switched RAM from $27EC to $94A7. The Sequencer also uses a few locations in Page Zero, Page Three and from $94A8 to $95F4 for variables, tables and bank switching routines.

Bank 1 of the RAM card, which consists of 4096 bytes, is used for ASCII tables and print routines accessed from the assembly language routines, which reside in Bank 2 of the RAM card. The upper 2K or so of RAM in Bank 2 is designated as an overlay area in which any one of many programs can reside.

PAGE ZERO USAGE:
The Sequencer uses many of the Apple system locations in Page Zero. Locations reserved exclusively for the Sequencer are the following:

06 DATA - the location used to hold the byte that is transferred to and from the Chroma and Expander ports.

07 IWRMSK - the interrupt mask image used to control interrupts from the Chroma port.

08,09 CTDNOF - a general purpose zero page register.

18,19 SIPTR - the input pointer to actual sequence data.

1A,1B SOPTR - the output pointer to actual sequence data.

1C,1D MAXLN - the maximum address of the sequence queue.

1E,1F MANSEQ - the start of the sequence file, 512 bytes before actual sequence data.

PAGE THREE USAGE:
Page three of the address space is used for variables accessed by BASIC. It is also used for bank switching routines that must reside in non-switched RAM. Details of the memory loacations and their functions is given in APPENDIX C, Program Overlay Structure and User Programs, in the copy of OVERLAY EQUATES.

SEQUENCE RAM:
The sequence file resides from $27EC to $94A7 non-switched RAM. The first two pages contain variables and tables that are to be stored with the sequence. Actual sequence data resides in a circular FIFO buffer (queue) starting at location $29EC and ending anywhere up to $94A7. The process of playing or recording restores the data at the previous end so, after any operation, the data can straddle the end of the buffer and wrap around to the beginning. The beginning and ending of the actual sequence data is marked with an EOS command (hex 00).

The sequence buffer can easily be relocated and its length defined by changing a few BASIC lines in the file CHROMA.BEGIN, which is the program that is first run upon booting the system. CHROMA.BEGIN sets up a few variables and determines what instruments are on line, then it runs RECORD TRACK.1, the Sequencer controlling program. When the Sequencer loads a sequence from disk, it automatically relocates it and checks to see that there is enough memory. When a sequence is saved to disk, the sections of memory (up to three possible) that contain the sequence are oriented into a continuous NEW TYPE A file. The structure of this file and the details of the variables and tables stored with the sequence are explained in APPENDIX B, Sequence/Program Data Structure.

ASSEMBLY ROUTINES:
Most of the actual work is done by a collection of assembly language routines that reside in the RAM card. The assembly routines are accessed from BASIC by setting a pointer CALNUM and calling a routine BASCAL on page three to switch the RAM and jump to the routine. CALNUM points to the assembly routine address in a table CALLS that resides in non-switched RAM from $94A8 to $94F4. The Apple ROM routines (both Monitor and BASIC Interpreter) are accessed from the assembly routines by a software multiplexer in Bank 2 that sets a pointer MNLOC then calls a demultiplexing and bank switching routine in page three called ROMSUB. Interrupts are handled essentially the same way. In all of these cases, the status of the RAM card is kept in a register called RAMSTAT and the card is restored just before the RTS or RTI is executed.

OVERLAY ROUTINES:
For a complete description of Overlay routines, where they reside, how they are accessed, etc. see APPENDIX C, Program Overlay Structure and User Programs.

GETTING MORE NOTES OUT OF THE SYSTEM

No matter how much memory a system has available for storage, there will be a need for more. It was with this in mind that we have provided two memory saving functions to be performed after most of a sequence has been recorded. One function reduces timing resolution and the other deletes measure commands. Both of these functions remove data that may still be valuable if you intend to record more than about one more track.

^Y-SCRUNCH	Reduces timing resolution from two time values (1 time value if using an External or Single Step timer source) to six time values. These are not TIME INCREMENTS, but actual time values. This may result in fewer 3 byte time commands stored in the sequence file.
^X-DELETE MEASURES	Removes all 3 byte measure commands and resets the sequence to as if there was never a click track. This eliminates the possibility of setting endpoints and makes it difficult to make the sequence loop in time. This function can only save a few notes worth of memory, so think about what you are giving up for it.

MEMORY SAVING TECHNIQUES:

The Chroma outputs Performance Control information in duplicate whenever a linked program is used. It doesn't matter if it's link upper, lower or unison. If the linked program does not make use of these performance controls then it is a waste of memory space when you record. To maximize the use of available memory space it is wise to be aware of this situation when it occurs and record the two tracks separately.

The Chroma will output all notes that are pressed, even if the voice is monophonic. In LINK UNISON mode, all attacks, releases and performance control changes are duplicated as instrument 0 and instrument 1 commands. Therefore, if you are recording a LINK UNISON program and the LINK is a monophonic program, you are wasting memory. It would be better to record each track separately and play one note at a time with the monophonic program.

The Sequencer stores a 3 byte time command before an event command if that event occurred more than 2 timer states after the previous event (1 timer state if using an external or single step clock). If you play your chords more precisely and record subsequent tracks more accurately, the Sequencer will use less memory space.

The Pressure Sensor Option will work with the Sequencer when it becomes available for your Chroma. The amount of data generated by the pressure sensor is analagous to a separate pitch bend lever for each key. Therefore, memory will be used up very quickly when recording with pressure. It is for this reason that we have elected to keep the "pressure switch" separate from the normal record mode, as indicated in the main menu.

INTERFACING TO DRUM MACHINES AND OTHER SEQUENCERS

The Sequencer can be controlled by external timing information in many different ways. The most important Sequencer operation parameters when trying to interface it to another system are the TIMER SOURCE and TIME INCREMENT. The SYNC input can also be used to help accomplish this. The details of these parameters, including checking the SYNC input, are presented in Chapter 9, RECONFIGURE and will not be duplicated here. In this APPENDIX, it will be assumed that you understand these parameters.

We cannot say that the Sequencer will interface to particular drum machines because we have not tried them. We can, however, make suggestions as to how one might connect and operate such an interface. We are currently evaluating particular drum machines for compatibility and will keep you updated with our results by mail (if you send in your warranty registration card).

THE EXTERNAL TIMING OPTIONS

EXTERNAL CLOCK INPUT:

The Sequencer does not measure the EXTERNAL clock input, but uses it as a source of interrupt. This means that a relatively high frequency is required for enough resolution in real time multi-track recording, which is what the Sequencer does best.

The EXTERNAL clock input specifications are detailed in APPENDIX O. Essentially, the circuitry is a zero crossing detector and can operate with a wide variety of input levels and waveforms. The clocking is accomplished on the falling edge of the clock input. You can probably connect either the metronome or external clock output of a drum machine to the EXTERNAL clock input of the Sequencer.

In general, there is a slight resolution problem due to the low external clock frequencies of most drum machines on the market. They are different between units, but generally the frequency is 24 or 48 pulses per beat. The speed of the drum sequence can generally be varied from 40 to 250 BPM so the clock frequency is between 16 and 200 Hz. This is enough resolution for sequences that contain only a few tracks; beyond that, you will begin to notice timing errors. Some drum machines output clocks up to 400 Hz, which should present little or no problems.

With the right recording techniques, however, this timing quantization error may be used to provide "error correcting". In other words, depending on the frequency, it may force notes to occur at 1/32 notes, etc. Release command times will probably present a problem unless corrected by using a TIME INCREMENT of greater than 1X while recording and changing some release times in the Editor. An alternate method is making sure the clock rate is at least equal to 1/32 if you want to play 1/8 notes, etc. The attacks will most likely be on time and the note will last at least the equivalent of a 1/16 note.

SINGLE STEP:

Lower frequencies (300 Hz to DC) can generally be accomodated using a Single Step2 TIMER SOURCE. The Single Step2 timer source mode accepts a TTL compatible signal connected to the SYNC input. This input can also be used with a footswitch to manually increment the timer during record or playback mode.

The general procedure would be to record using a slow clock or footswitch in Single Step2 mode, then play back using the EXTERNAL clock mode. This can be cumbersome, but it does work. In one experiment, we have used this method and clocked the Sequencer to play with an envelope follower driven by the bass drum output of a drum machine. In the studio, the control voltage output of a VCA (such as that found in a KEPEX) can be used to clock the Sequencer. Generally, you want a short attack and release time, maximum ratio and adjust the threshold for whatever drum you are using for the clocking.

USING THE SYNC INPUT:

If you record with an external or single step timer source, you will need to tell the Sequencer when to start playing a sequence. This is required because the Sequencer normally sets its internal timer to the first time value in the sequence before playing. Therefore, the sequence will immediately begin to play when you select **P**. You do not want this when using a drum machine. Instead, you will select **P** and want the Sequencer to wait for the drum sequence before it starts playing.

This can be done by one of two methods. One way is to set up the Sequencer to wait for a positive going pulse on the SYNC input (see Chapter 9, RECONFIGURE). The other way is to INSERT a TIME 0 command at the beginning of the sequence via the Editor.

If you use the SYNC method, the signal must be TTL compatible. Generally, you can use the external clock of a drum machine if you find a way to split the signal. This can be done with an ordinary Y adapter. When the Sequencer starts getting the clock pulses from the drum machine, signifying that it is playing, the Sequencer will start to play its sequence.

When using the sync method, you must consider what happens in loop mode. When looping back, the sequencer does not check the SYNC input. Instead, it will loop back immediately (assuming you don't have a click track). The pressing of the footswitch or space bar at the end of recording is critical in this application. The loop processing time may also cause cumulative error problems if the external clock speed is relatively high.

The method that requires INSERTING a TIME 0 command in the editor works very well except in loop mode. The amount of time between 0 and the first time value will be heard as a rest every time the sequence loops back. If your sequence can accomodate this, we recommend it over the SYNC input method described above.

This is how you do it: When in the Editor, notice the first time value. Then CHANGE that time value to 0. Then INSERT a TIME command with the old time value right after the DEFINE command. When you select **P** from the main menu, the Sequencer will DEFINE the track then wait for the first clock pulse from the drum machine.

APPENDIX G - SLOT INDEPENDENCY

SLOT INDEPENDENCY - HOW IT WORKS

In the Apple, there are 16 memory mapped I/O locations and 8 scratch pad memory locations for each slot. There is also a 256 byte address space allocated to each slot. The addresses of all these locations are different for each slot and are related to the slot number. The Sequencer only uses the I/O locations, so our discussion will center around addressing those 16 locations.

The I/O locations take the following form:
$C0XX where XX is $80 plus the (slot number times 16) plus $00-$0F for 16 individual locations.

Normally, one would use $C080-$C08F as a base, and index with the slot times 16 to address the desired location. There are problems with this approach if the programmer wants to use the X and Y registers for something else, especially if he wants to load the registers with what is in the I/O locations. The Sequencer, therefore, takes a different approach as outlined below.

The Sequencer uses 'self modifying' code to address these I/O locations. Initially the program is assembled with the following instructions when addressing the I/O:

```
BRK             does a software interrupt
.DA (00-0F)     location of I/O relative to the first
.DA (opcode)    the operation (LDA,LDX, etc.)
```

When the section of code is first executed, the BRK causes the processor to jump to a routine in which the previously described section of code is changed to thereafter perform the desired operation. This sounds time consumming, but it only occurs once for each reference after the power is turned on.

For an example, if we wanted to load the X register with the pedal value from the ADC and the Interface PC board was in slot 5, the initial object code would be:

```
00  BRK
02  location of RDADC
AE  LDX absolute
```

After the BRK routine the object code would be:

```
AE LDX absolute
D2 addr lo of RDADC
C0 addr hi of RDADC
```

The BRK routine then returns to the modified code, where execution continues. The next time this section of code is referenced, there is no BRK instruction, so execution is straight foreward.

The Assembly Language Code used to do this is a "macro", which takes the form of LDIO opcode,I/O label, as described in the OVERLAY.EQUATES file in APPENDIX C.

A description of the BRK routine itself follows. When the 6502 enters a break routine, the stack contains the Program Counter +2 of the location of the BRK instruction. It, therefore, points to the desired opcode. The flow of the BRK routine is as follows:

- decrement the PC (on the stack) by two
- get the opcode and store it at PC
- get the relative I/O location
- add the slot times 16 (IFSLOT)
- store the I/O location low byte at PC+1
- store $C0 at PC+2
- return from INTERRUPT

WHAT TO DO IF YOU NEED SLOT 5 FOR SOMETHING ELSE

Some peripherals for the Apple II are slot dependent, which means they must be installed in a particular slot. The Interface PC card is **NOT** such a peripheral. You must, however, first power-up the system with the card in slot 5. If you need slot 5 for something else, take that something else out of slot 5 temporarily and perform the steps outlined in the Installation section then do the following:

1. Select 'X' (RECONFIGURE) from the main menu. After the Sequencer loads that overlay, it will print another small menu. To thoroughly understand RECONFIGURE, you should read Chapter 9. It is not necessary at this time, however, if you carefully follow these instructions.

2. Select 'C' (CHANGE SET UP) from this RECONFIGURE menu.

3. Keep typing <RET>, which means DEFAULT, until it tells you that the Interface Slot is 5. Instead of typing <RET>, type the number of the slot that you want to use for the Interface PC card **THEN** type <RET>. After that, keep typing <RET> until you get back to the RECONFIGURE menu.

4. At this point, type 'V' for VIEW CURRENT STATUS and check to see that the Interface Slot is truly set up for the slot you intended. After you are satisfied that this is the case, notice the disk controller slot number for future reference and type any key to get back to the RECONFIGURE menu. If you are not satisfied then type any key to get back to the RECONFIGURE menu and goto step 2.

5. Type 'S' for SAVE SET UP. When it asks you for the name of the set-up file carefully type "PAGE 3.HEX" then <RET>. Make sure you do **NOT** include the quotation marks and that you **DO** type a space between PAGE and 3.

6. After this is done, you are back at the RECONFIGURE menu so type <RET> to exit into the main menu.

7. **IMPORTANT!** The first and only thing you should do at this point is type <ESC> to escape into APPLESOFT. After you get the normal APPLESOFT prompt (]) type PR#X where 'X' is the number of the slot that the DISK CONTROLLER card is in. This will reboot the system and load in your changed configuration. Any time you power up from now on the system will expect to see the Interface PC card in the slot you said it would be in. If you have any questions regarding the RECONFIGURE procedure, see Chapter 9.

BOARDS, CHANNELS, INSTRUMENTS and TRACKS

The Chroma and the Apple Sequencer can produce very complex sequences if it is used with a knowledge of the Chroma architecture. It is preferred to view the Chroma as an eight channel synthesizer with provisions for 16 channels within certain limitations. Each of the eight channels has two oscillators, two filters and two amplifiers. .Sixteen channels are available only when the program patch parameter is 0.

The Interface Command Set supports the eight channel concept. The three least significant bits of certain command opcodes denote instruments. Instruments are numbered from 0 to 7, with the Chroma keyboard normally using instrument 0. The keyboard is also assigned instrument 1 if the current program has a link. The way the Sequencer records is that it inputs Instrument 0 and 1 commands from the Chroma and assigns track numbers. These track numbers, in turn, are assigned available instrument numbers when playing the sequence. The sequence can, therefore, play up to eight tracks at a time.

Due to the interface command opcode arrangement, it is possible to transform the opcode before storage and use the four least significant bits for the track designation. The Sequencer currently, however, always stores instrument 0 opcodes and a separate track byte. The Sequencer currently supports 16 tracks total.

Throughout this manual, tracks and instruments are used interchangably unless noted. Tracks refer to instruments defined by the Sequencer.

It is important to understand how the Chroma allocates channels and boards among instruments and what happens when the user plays more notes than can be sounded with the number of channels available for that instrument. Channel and board allocation for each track depends on the number of tracks currently defined and the Keyboard Algorithm and patch parameter of the programs that are defined by the tracks.

Monophonic instruments are always assigned one channel and one board. If the patch parameter for the program is 0, then only one oscillator/filter/amplifier on that board is used. If the patch parameter is non-zero then all hardware on the board is used.

The number of channels and boards assigned to polyphonic instruments depend upon the number available at any instant. For the purposes of initial explanation, let us assume that all programs, monophonic and polyphonic have a patch of non-zero. This is reasonable since most people prefer the sound capabilities of those patch configurations. A single polyphonic track, just as an unlinked passage played from the keyboard, is assigned all eight channels and uses all the hardware on each board. If there are two polyphonic tracks, each is assigned four channels (and four boards). If there are an odd number of polyphonic tracks, then the Chroma assigns channels as evenly as possible but gives priority to the lowest instrument numbers.

In the case of the Sequencer, this means that the tracks that start earlier are given more channels if the total number of tracks is odd. For instance, if three polyphonic tracks defined, the first two have three channels and the third one has two channels.

The Sequencer marks the beginning of a track with a define command and the ending with an undefine command. Keep in mind that the first track may not be track #1, since another track can be recorded with its first note occurring before track #1 starts. Instruments are assigned in the order of define command occurrance. Every time a define or undefine command occurs, channels are reallocated. If all tracks are defined as polyphonic, then the number of channels allocated to that instrument are likely to change during reallocation. If a track is monophonic, it will not change because monophonic instruments are always assigned one channel.

This is a good place to explain what happens when the musician or the Sequencer tells the Chroma to play more boards than it has available. If you play the Chroma keyboard you will notice that if the program patch is nonzero and you try to play more than eight notes, then the Chroma will release the first (oldest) note played. If you do the same thing with a program that is linked lower with another program and both programs have a patch parameter of nonzero then you can play only four notes on either side of the split before the Chroma "steals boards." The Chroma will not steal boards between instruments. The same thing occurs with the Sequencer, but the effect is much more pronounced since many more instruments are defined and each instrument ends up with fewer channels. When channels are reallocated and the musician or Sequencer asks for more notes in that track than is possible then the Chroma releases the oldest note in that track.

If you digested the above description of channel allocation, you must realize how important it is to know before hand how many tracks you will record and whether these tracks will be polyphonic or monophonic.

To make the situation a little more complex, consider tracks that are defined as programs that have patch parameters of zero. This means that each note uses one oscillator/filter/amplifier. If the Keyboard Algorithm is monophonic, one half of the board is wasted. If the program is polyphonic and only one instrument is defined, then you can play 16 notes before the Chroma steals boards. If there are three instruments defined and the first is polyphonic with patch 0, the second is polyphonic with patch nonzero and the third is monophonic with patch 0 or nonzero then the first instrument can play eight (two times four) notes, the second can play three and the third can play one. If you play more than eight notes in the first instrument but only one in the second instrument, the Chroma cannot use channels allocated to the second instrument and will release the oldest notes played by the first instrument.

CHROMA COMMAND DESCRIPTIONS

The computer that is connected to the Chroma via the interface cable communicates with the Chroma by sending and receiving commands. A command consists of:

A byte that specifies the command. If the command applies to one of the eight "instruments" within the Chroma, the instrument number will be encoded in this byte, too.

Zero or more bytes that specify parameters of the command. Although most commands require specific numbers of parameters, a few commands are variable in length.

Certain conventions are adhered to in the command language:

Undefined commands are considered to be No Operation commands; that is, undefined commands are ignored. All No Operation commands have no parameters.

Command code zero and command code FF (hex) will always be No Operation commands, even for future instruments that utilize this interface.

Command code 1 will always be an Identification command, for this and any other instrument utilizing this interface.

If a two-byte quantity (such as a memory address) is to be transferred, it will be sent most significant byte first, just the way you would write it on paper.

If a command is variable in length, the second byte of the command will specify the variable number of data bytes. This is not the same as the length of the command, as the count does not include the command code, the length byte, or any other fixed parameters for the command. The Peek command is a good example of this.

If a command is variable in length, the second byte of the command will specify the length as follows: values 1 to 255 represent byte counts of 1 to 255, and a value of zero represents a byte count of 256.

Any command that could conceivably "crash" the Chroma through misuse will not be allowed until a special "unlock" command is first issued. This minimizes the chance of a crash if the Chroma should receive garbage from a malfunctioning computer.

The commands fall roughly into three categories, according to protocol:

There are those commands that are issued by the controlling device and processed by the Chroma with no response.

There are those commands that are issued by the controlling device and require a specific response from the Chroma. The response will always be a "command" starting with the same code that was received from the controlling device.

There are those commands that establish modes within the Chroma that allow the Chroma to subsequently transmit unsolicited "commands" when certain events occur. The unsolicited commands will generally look like commands from the first group above.

The command set can also be split into two categories, according to destination:

There are those commands that are addressed to the Chroma as a whole. The lower command codes are assigned to these commands.

There are those commands that are addressed to individual instruments within the Chroma. The higher command codes are assigned to these commands. The three least significant bits of these command codes hold the instrument number.

What follows is a complete description of each command, along with the numerical code (in hexadecimal) for each command byte.

No Operation 00

The only significance of this particular No Operation (as opposed to any of the undefined command codes) is that the Chroma sends this code upon power-up or reset.

Identification 01

The Chroma (or any future instrument) will respond with three bytes, an Identification command, a device code (1 for the Chroma), and a software revision level code.

Read Program 02 pp

The Chroma will respond by transmitting program number pp. The information is transmitted as a Read Program command and 59 data bytes. (If pp is not between 0 and 50, the data bytes will be undefined.)

Write Program 03 pp dd ... dd

The 59 data bytes dd ... dd are written into program number pp in the Chroma. (If pp is not between 0 and 50, the data will be accepted and ignored.)

Load Packet 04

One packet of information is read from the cassette interface, its error detection codes are checked, and the result will be returned via the interface in the form:

04 nn dd ... dd

nn specifies the number of data bytes in the packet, and the dd bytes are the contents of the packet. The first byte of the packet (the first dd byte) is always the packet ID, which identifies the type of packet. The packet ID for valid data is always non-zero. If an error occurs in the reading of the cassette, a special error packet with an ID of 0 is returned.

This command starts reading from the cassette immediately. This can cause a problem if the cassette was previously idle. See the Tape Space command below.

The types of packets that are currently defined, and the forms the Chroma return them in, include:

Error Packet 04 02 00 nn

The length is 2, the ID is 0, and nn will be 0 if a read error is detected or FF hex if the cassette was not running (or was shut off in mid-operation).

Program Packet 04 3C 01 dd ... dd

The length is 60 (3C hex), the ID is 1, and the 59 bytes of data represent a Chroma program.

Program Number Packet 04 02 02 nn

The length is 2, the ID is 2, and the single byte of data consists of a valid program number (0 to 50). This type of packet appears, with a program number of 1, at the beginning of a tape recorded with SAVE ALL.

Stop Packet 04 01 03

The length is 1, the ID is 3, and there is no data in the packet. This type of packet appears at the end of a tape recorded with SAVE ALL.

Save Packet 05 nn dd ... dd

The packet dd ... dd containing nn bytes is written to the cassette. The first dd byte, which is the packet ID must be non-zero. The Chroma will respond when the operation is complete with 05 00 if the operation completes normally or 05 FF if the cassette isn't running.

Read Parameter 06 pp nn

Parameter number nn in program number pp is read and returned in the form 06 vv, where vv is the parameter value. If pp is not between 0 and 50, or if nn is not between 0 and 100, the vv value will be undefined.

Write Parameter 07 pp nn vv

Parameter number nn in program number pp is set to value vv. If pp is not between 0 and 50, or if nn is not between 0 and 100, the vv value will be ignored. If the vv value is not within the range defined for the parameter, the result is undefined, except that the parameter will never be set to an illegal value.

Panel Switch Off 08

The "panel switch" referred to is the software switch which "connects" the Chroma panel to the interface. When the Chroma receives this, it will echo it and disconnect the panel from the interface.

Panel Switch On 09

When the Chroma receives this, it will echo it and connect the panel to the interface. While this mode is in effect, the Chroma will transmit certain commands when the following events occur:

Whenever a program is selected, a Define command will be transmitted for instrument 0 and either a Define or an Undefine command will be transmitted for instrument 1, depending upon the existence of a link.

Whenever a parameter is changed, a Set Parameter command will be transmitted for instrument 0.

Whenever the link balance is varied, Volume commands will be transmitted for instruments 0 and 1.

Performance Switch Off 0A

The "performance switch" referred to is the software switch that "connects" the various performance controls to the interface. When the Chroma receives this command, it echoes it and disconnects the performance controls from the interface.

Performance Switch On 0B

When the Chroma receives this, it will echo it and connect the performance controls to the interface. While this mode is in effect, the Chroma will transmit certain commands when the following events occur:

Whenever a key is pressed on the keyboard, an Attack command will be transmitted for instrument 0, 1 or both, depending upon the link mode and keyboard split.

Whenever a key is released on the keyboard, a Release command will be transmitted for instrument 0, 1 or both, depending upon the link mode and whether or not an attack had already been sent for the note.

Whenever a lever, pedal or footswitch moves, the appropriate command is transmitted for instrument 0, and for instrument 1 if a link is in effect.

Peek 0C aa aa nn

The Chroma responds by transmitting nn bytes from its internal memory starting at location aaaa. The response is in the form:

0C nn dd ... dd

where the dd bytes are data bytes from ascending addresses.

Peek Two Bytes 0D aa aa

The Chroma responds by transmitting two bytes from its internal memory at locations aaaa and aaaa+1. The response is in the form:

0D dd dd

This command is guaranteed to extract the two bytes concurrently, with no chance that the memory locations could be altered between the transmittal of each byte.

Poke 0E aa aa nn dd ... dd

The nn data bytes dd ... dd are poked into the computer's address space starting at location aaaa. If an Unlock command has not been issued since the Chroma was powered up (or reset), the entire command will be read in and ignored.

Poke Two Bytes 0F aa aa dd dd

The two data bytes dd dd are poked into the computer's address space in locations aaaa and aaaa+1, respectively. If an Unlock command has not been issued since the Chroma was powered up (or reset), the entire command will be read in and ignored. This command is guaranteed to poke the two bytes concurrently, without danger of the computer utilizing half of the old contents and half of the new contents.

Tap Panel 10

The panel tapper is triggered, unless it has been disabled.

Unlock 11 00 FF

This sequence must be transmitted in order to enable the Poke and Poke Two Bytes commands.

Tape Space 12

The cassette motor will be run for two seconds. Upon completion, the Chroma will respond with 12 00 if the cassette was running, or 12 FF if it was shut off.

The purpose of this command is to allow startup time before other cassette operations. If a sequence of Save Packet commands are to be issued, they should be preceded by **two** Tape Space commands. In addition, if the packets are to be individually readable, they should be separated by two Tape Space commands. A **single** Tape Space command should be issued prior to a sequence of Load Packet commands.

Restore 13

The Chroma is restored to the state reflected by its panel settings. All instruments are undefined except instrument 0 and possibly 1, which are set up according to the currently selected program. The panel switch, performance switch and pressure switch are turned off, and a Panel Switch Off, Performance Switch Off and Pressure Switch Off command are echoed, in that order.

Pressure Switch Off 14

The "pressure switch" referred to is the software switch that "connects" the keyboard pressure sensors to the interface. When the Chroma receives this command, it echoes it and disconnects the pressure sensors from the interface.

Pressure 68+i kk pp

Instrument i is told to set the key pressure input for note kk to value pp. The pressure is an unsigned number from 0 to 63.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the pressure switch is on and the measured pressure on a depressed key changes. Pressure commands only occur between the corresponding Attack and Release commands for the same note.

Information 70+i

The Chroma responds by echoing the command and sending four information bytes. Currently, only the first byte is utilized, and contains the number of channel boards assigned to instrument i. The other three bytes are zero.

Volume 78+i vv

The Chroma sets the volume of instrument i to vv. The value vv is a linear control from 0 to 255, and is nominally 255. Thus, to reduce the volume of an instrument 6db, the correct vv value would be 128.

This command will be transmitted (for instruments 0 and 1) by the Chroma if the panel switch is on and the Link Balance parameter is varied.

Lever 1 80+i vv
Lever 2 88+i vv

The Chroma sets the value of the appropriate lever input on instrument i to vv, where vv is a signed 2's complement byte in the range -128 to +127. This range corresponds to the mechanical range from "pull" to "push", with 0 corresponding to "at rest".

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer moves a lever.

Pedal 1 90+i vv
Pedal 2 98+i vv

The Chroma sets the value of the appropriate pedal input on instrument i to vv, where vv is a number in the range 0 to 255. This range corresponds to the mechanical range from "heel" to "toe".

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer moves a pedal.

Footswitch 1 Down A0+i
Footswitch 1 Up A8+i
Footswitch 2 Down B0+i
Footswitch 2 Up B8+i

These commands activate or deactivate the footswitch functions on instrument i.

These commands will be transmitted (for instruments 0 and possibly 1) by the Chroma if the performance switch is on and the performer presses or releases either footswitch.

Define C0+i pp aa bb cc dd ee ff

Instrument i is defined according to program pp (which must be in the range 0 to 50). The remaining bytes specify initial values for the performance inputs:

aa: lever 1	bb: lever 2
cc: pedal 1	dd: pedal 2
ee: volume	ff: footswitches

The footswitch byte uses the most significant bit to represent footswitch 1 and the next most significant bit to represent footswitch 2. A 0 means up, 1 means down. If pp is not between 0 and 50, the Chroma will not define the instrument according to garbage data, but nothing more is promised.

This command causes channel boards to be reallocated as fairly as possible among defined instruments. If this command requires that one or more channel boards be

robbed from another instrument, the computer will be kind enough to try and pick boards that aren't currently sounding.

This command will be transmitted (for instrument 0 and possibly 1) by the Chroma if the panel switch is on and the performer selects a program or a link. Although instrument 0 is internally defined by program 0, the Define command that is transmitted whenever a program is selected includes the current program number as shown in the 2-digit display.

Undefine C8+i

Instrument i is removed from operation, and any channel boards assigned to it are redistributed among any other instruments.

This command will be transmitted for instrument 1 by the Chroma if the panel switch is on and an unlinked program is selected or a link is cleared.

Attack D0+i kk vv pp

Instrument i is told to attack note kk with a velocity vv and an initial pressure pp. The key number is a signed, 2's complement byte that must be in the range -64 to +63. The Chroma's keyboard has a range from -32 to +32, with 0 being middle C. The velocity must be a number from 0 (softest strike) to 31 (hardest strike), and the pressure must be a number from 0 (no pressure) to 63 (full pressure).

The result of this command depends upon the keyboard algorithm parameter in the program that the instrument is defined by.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the performance switch is on and the performer presses a key.

Release D8+i kk vv

Instrument i is told to release note kk with a velocity vv. The result of this command depends upon the keyboard algorithm parameter in the program that the instrument is defined by.

This command will be transmitted for instrument 0 and/or 1 by the Chroma if the performance switch is on and the performer releases a key.

Set Parameter E0+i nn vv

Instrument i temporarily sets parameter nn to value vv. This does not affect the setting stored in non-volatile memory, which means that it won't affect other instruments defined according to the same program and it won't affect this instrument if it is redefined according to the same program. Only those parameters that pertain to the tone generation may be set with this command. These include:

1 through 5: control parameters
6 through 50: A parameters
55 through 100: B parameters

Any other parameter number will cause the command to be ignored. If vv is not within the valid range for the selected parameter, the only guarantee is that the parameter will not be set to an illegal value.

This command will be transmitted for instrument 0 by the Chroma if the panel switch is on and the performer varies one of the parameters.

Status E8+i

This command causes the Chroma to respond with:

E8+i pp aa bb cc dd ee ff

where the seven parameters represent the same quantities as the parameters of the Define command. If the instrument is undefined, the program number returned will be FF and the remaining bytes will be undefined. If the program number is 0, the program number in the display will be used instead.

Squelch F0+i kk

Any channels in instrument i that are assigned to key k are squelched by setting their envelopes to 0. This doesn't affect the channel assignment tables. Even latched channels may be squelched. If kk is -128 (80 hex) all channels will be squelched.

I/O LOCATIONS AND THEIR FUNCTIONS

NOTE: assumes slot 5 for addresses

ADDRESS	OPERATION	NAME	DESCRIPTION
C0D0	Write	DISEXTC	Disable the External clock
	Read	ENEXTC	Enable the External clock
C0D1	Write	------	Not used
	Read	------	Not used
C0D2	Write	STRADC	Start the ADC conversion
	Read	RDADC	Read the ADC value
C0D3	Write	CLRTIM	Clear the Timer Interrupt
	Read	RDMISC	(7) Time - LO indicates time Int. (6) XXFULL - LO= expander data still on port. (5) FOOTSW - up(NO) = HI (4) Conversion Ready = LO
C0D4	Write	CLICK	Output HP filtered TTL click
	Read	------	Not used
C0D5	Write	XWRMSK	Write external interrupt mask: XIMASK NOT (7), XOMASK NOT (6).
	Read	RDFLAG	(7) XIFULL NOT (6) XOFULL (5) SYNC IN (open=HI) (4)IXFULL NOT (Expander Input Available)
C0D6	Write	WREXTO	Chroma output port
	Read	RDEXTX	Expander input port
C0D7	Write	WREXTX	Expander output port
	Read	RDEXTI	Chroma input port

C0D8-C0DF NOT DECODED

A Bug In The Chroma Described

In Chroma software revisions up to and including REV 12 there is a bug that involves instrument numbers sent over the interface when program 0 has a link. When a linked program is selected for recording and any Lever or Pedal is moved, the Chroma sends over two lever or pedal commands which should be for instrument 0 and 1, but are always instrument 0. This bug will be corrected in REV 13 of Chroma software.

The bug will be noticed when you record things like pitch bends or vibrato in a link program. The linked program will fail to be modified when the sequence is played back.

Also, varying the link balance control when the panel switch is on will cause the Chroma to send volume commands for both instrument0 and instrument1 to indicate the absolute volume relationship between the instruments. The Chroma does send both volume commands but the value is always the same and that value is random bearing no relationship to the actual volumes of the instruments. This bug is also corrected in REV 13 software.

It is recommended, therefore, that users of the Computer Interface Kit upgrade their Chromas with REV 13 ROMs whenever they become available. REV 13 will also support the Pressure Sensor Option.

THE INTERFACE TEST PROGRAM

The USER UTILITY 20 program overlay is a small test program designed to allow the user to determine that his hardware is functioning properly. This program will also be used in the production of the Interface Kits at Woburn until I have time to design a more comprehensive test set/program.

To access the program, type ^V for the USER UTILITY bank 2, then type 0 for user number 0. The program will then be loaded and an initialization procedure will be ran. The display will indicate the status of the two ports. If the Sequencer beeps and tells you that either port is not responding then you either have a problem with that port or no instrument is connected.

The Tester is a command oriented program. Below is a short description of each command and the subcommands allowed within each operation:

^I - REINITIALIZE - Will attempt to send 260 NOPS to the Chroma and Expander port. Will initialize the output buffer for the OUTPUT commands described below. Will display the status of the two ports.

^T - TIMER TEST - Will initialize the timer registers to zero and display the counts as they occurr. The timer source (internal, external, or single step) and timer increment will reflect the current set-up. Typing a space will pause the display. Typing an R will reset the timer registers to zero. Typing a return will terminate the test. When using an internal or high frequency external (1 KHz or higher) timer source, a number that is missing in the sequence of values may or may not indicate a missing timer pulse.

^A - ADC TEST - Will read the ADC and display the values. This is useful for setting the range trimpots on the PC board. Typing a space will pause the display. Typing a return will terminate the test.

^K - CLICK TEST - Will output a click of 4/4 time at approximately 120 BPM. Typing a space will pause the output sequence. Typing a return will terminate the test. This test takes TIMSRC and TIMINC into consideration in determining the final output frequency of the click.

^F - FOOTSW TEST - Will display the current state of the footswitch and any change in that state. Will reflect the FOOTCK register in the current Reconfigure set-up. Typing return will terminate the test. The footswitch input is not debounced on release, so you may see UP DOWN UP when you release the footswitch during this test.

^S - SYNC CHECK - Will display the current state of the SYNC input and any change in that state. **Will not** reflect the current Reconfigure set-up. In other words, this test will always check the SYNC input and will only wait for one state change. Typing return will terminate the test. Since the SYNC input is not debounced, mutiple states may be displayed when a footswitch is used.

^C - CHROMA TEST - Will output bytes to the Chroma port and display the data received from the Chroma in response to the output bytes. The sub-commands allowed are:

^D Delete last entry
^A Abort output
^M (RET) output the buffer
any hex number (2 ascii chars) will be put in the output buffer (256 bytes maximum)

Good tests for the Chroma port are outputting the following commands (and inspecting the response):

```
OUTPUT -     ID : 01
INPUT  -        : 01 01 (Chroma) 02 (REV 2)


OUTPUT - WRPRG0 : 03 00 11 09 18 0F D7 F0
                  0A 00 F8 13 01 00 B0 7B
                  0C 00 FC FE 01 90 3D 80
                  C2 B5 89 97 36 00 00 01
                  C3 08 04 0F CB F0 00 00
                  18 0B 01 00 F8 7B 0C 00
                  7D 7E 01 90 3D 80 C2 AB
                  87 97 36 00 00
INPUT  -        : No response from Chroma
```

```
OUTPUT - RDPRG0 : 02 00 (Read Prog 0)
INPUT  -        : 02 11 09 18 0F D7 F0 0A
                  00 F8 13 01 00 B0 7B 0C
                  00 FC FE 01 90 3D 80 C2
                  B5 89 97 36 00 00 01 C3
                  08 04 0F CB F0 00 00 18
                  0B 01 00 F8 7B 0C 00 7D
                  7E 01 90 3D 80 C2 AB 87
                  97 36 00 00
```

^X - EXPANDER TEST - Same as above, but will test the Exander port.

^E - EXIT TEST - Will exit the Interface Test Program and return to the main menu.

COPYING DISKETTES

It is occassionally necessary to copy diskettes for backups, etc. It is principally for this reason that we have not copy protected the disks in the Sequencer system. We, in fact, encourage you to immediately make a back-up of the diskettes and use the back-ups. Store the originals in a safe, cool and dry place.

We recommend that you use the FID program instead of the normal COPY program for making these copies. For some reason the COPY program will not always work with the assembly source, program group and sequence files. The FID utility has no problem with these files.

MAKING NEW SEQUENCE DATA DISKETTES

The sequence data diskettes for 2 drive systems are very easy to make. Creating a new data diskette for a single drive system is relatively more involved. The procedures required are outlined below:

DUAL DRIVE SYSTEM:

(1) Exit into Applesoft by typing ESC from the main menu.

(2) Type LOAD HELLO,D2

(3) Take the original diskette out of drive two and replace it with the new unformatted diskette.

(4) Type INIT HELLO,D2

(5) LABEL the new diskette.

SINGLE DRIVE SYSTEM:

(1) Exit into Applesoft by typing ESC from the main menu.

(2) Type LOAD HELLO

(3) Take the original diskette out of our drive and replace it with the new unformatted diskette.

(4) Type INIT HELLO

(5) Take the new diskette out of the drive and label it.

(6) Insert a 3.3 Master Diskette in the drive and type BRUN FID

(7) Select FID menu item 1 (copy files) and specify the source and destination slot and drive to your disk controller slot and drive 1. Set filename to = (all files). Set prompt to no prompting. Follow instructions from here, inserting source disk (old disk) then destination disk (new disk) until the whole disk is copied. You will need to tell the computer to write over the HELLO program that already exists on the new diskette. We suggest you read the Apple DOS Manual before you use FID.

(8) Put the new diskette in your drive and boot the system by typing PR#6 or whatever your slot number is.

(9) When you are in the main menu, type **D** for delete sequence and delete the sequences you do not want on this new diskette.

(10) Type **Q** for Program File Management, then select **3** to delete any program groups you do not want on this new diskette.

USING OTHER SYSTEMS WHILE THIS CARD IS IN PLACE

The way the hardware in the Interface PC board is currently set up may cause problems with other cards that use or inadvertently enable the interrupts. The Interface PC card never stops generating interrupt requests if the card is running on the internal clock. Whether the card operates from the external or the internal clock is random upon power-up. If the Interface card is installed and another system is booted that uses or enables the interrupts, our card can impede proper operation of that system. We recommend that you either remove the Interface PC card when not running our system or perform the following instructions before booting the other system:

(1) Turn on the Apple with the Master Diskette or other diskette that does not enable the interrupts.

(2) Remove any External Clock input from the Interface Connector Chassis.

(3) From Applesoft, PEEK (-16175) then POKE -16172,0.

(4) THEN boot the system you wish to run.

This enables the EXTERNAL CLOCK and clears the interrupt request line. If no clock signal is inputted then interrupts will not ocurr from the Interface PC card.

The user could modify the HELLO program in the system he wants to run to do this before it enabled the interrupts. The Apple powers up with interrupts disabled.

It is also possible to insert a SPST switch in line with the IRQ line on the Interface PC board. The switch could be mounted in the extra jack cutout in the connector box assembly. Performing the modification will void your warranty unless performed by an authorized Rhodes Chroma Service Center. Call (617) 938-1610 for the name of the one nearest you.

HARDWARE DESCRIPTION AND SPECIFICATIONS

A Schematic and component layout drawing of the Interface PC board is included in this section for your reference. It is included solely for the purposes of explanation and is not to be used for servicing your own board. Refer servicing to an authorized Rhodes Chroma Service Center or call the Rhodes Chroma Service Department at (617)938-1610.

Also, please obey the law regarding copyrights!!!

THE INTERFACE PC CARD:

The hardware of the Interface PC Board is divided into the following sections:

(1) I/O Address Decoder
(2) Chroma Port
(3) Expander Port
(4) Interrupt Control
(5) Click Output
(6) Clock Circuit
(7) Analog to Digital Converter
(8) Status Input Port

Interconnection is provided by:

- The Apple Slot edge connector, which allows communication with the Apple and supplies the power (approximately 160 milliamps from +5V, 60 milliamps from +12V and 40 milliamps from -12V). The slot connector pin numbers are represented by small rectangular blocks on the Schematic Diagram.

- The 8 pin phono jack connector (J7), which supplies the Click output and the Pedal, Footswitch, Sync and External Clock inputs.

- The two 26-pin port connectors J5, the Chroma port, and J6, the Expander port.

I/O ADDRESS DECODER:

The address decoder circuitry uses the DEVICE select, R/W, Phase 1 clock and three least significant address lines to decode the addresses of the various I/O functions of the Interface Card. The software locations of these functions are detailed in APPENDIX J and will not be repeated here. The outputs of Z10 and Z15 are active low enable pulses that are used to latch data and/or enable the reading of data from tri-state buffers. They are also used to provide control pulses for clearing the interrupt, starting the ADC, selecting the timer source and outputting a click track pulse.

CHROMA PORT:

The Chroma port consists of an 8-bit input port with handshaking and an 8-bit output port with handshaking. The port connects to a mirror image of itself in the Chroma; that is, each input port line (mnemonics starting with XI) connect to the corresponding output port line (mnemonics starting with XO) at the other end of the interface. The interface really only consists of the output latch (Z5), the input tri-state driver (Z7), the four NAND gates and an inverter. The two transistors Q3 and Q4 are for isolation when the power is shut off. All he other stuff is just for noise rejection. When the Sequencer wants to transmit a byte of data, it checks the XO FULL line by reading the Status Input Port to see if the last byte it sent has been received yet. When it has, it writes the byte into the latch with the WR EXTO strobe coming from the I/O Address Decoder. This causes the flip-flop consisting of gates Z11a and Z11b to be set, and pulls the XO FULL line low. This tells the Sequencer that the output port is full (and not to send any more data yet) and tells the other end of the interface that there is fresh data to be had. When the other end reads the data, it will pulse the acknowlege line XO ACK, which resets the flip-flop and resets XO FULL high (inactive) again. This tells the Sequencer that it can send another byte of data.

The input interface performs the other side of the same task. When data arrives from the other end via the XI lines, the XI FULL line will go low, telling the Sequencer that the fresh data has arrived. When it reads it, using the RD EXTI strobe, the acknowlege XI ACK will be pulsed, causing the flip-flop at the other end to be cleared, and causing XI FULL to go high (inactive) again.

The remaining gates Z11d and Z11c are used to allow masking of the XOFULL and XIFULL lines for interrupt control.

INTERRUPT CONTROL:

Normally, the Sequencer is ready to accept data from the interface, and the XI MASK line from Z16b is high (inactive). This means that an incoming byte, which is accompanied by XI FULL going low, will cause the input of Z12c to go low, turning on Q1 and interrupting the Apple. The only times the Sequencer activates XI MASK to prevent input interrupts is if the device at the other end of the interface is sending data faster than the Sequencer can process it.

Normally, the Sequencer has no data to transmit, and if it does, the interface is usually ready for it, as signified by a high (incative) XO FULL. If, however, the Sequencer has data to send and the output port is still full from the previous data transfer, the Sequencer will store the byte of data in a FIFO (first in first out) queue in its memory and set XO MASK high (inactive) from Z16a, thus unmasking output port interrupts. The output port interrupt occurs whenever the device at the other end of the interface gets around to reading the data off the interface and sending back an XO ACK pulse. Then the Sequencer will take time out from whatever it is doing to pull a byte from the end of the FIFO queue and output it. Only when the queue is empty does the main computer mask output interrupts again by setting XO MASK low (active).

Upon power-up, the XOMASK is set low (active) and XIMASK is set high (inactive) by the system RESET line.

Interrupts can also occur from the Clock Circuit. Interrupts from the Clock Circuit cannot be masked, see APPENDIX N.

The Sequencer determines the source of the interrupt by reading the XOFULL, XIFULL and TIME lines from the Status Input Port.

EXPANDER PORT:

The Expander Port works exactly like the Chroma Port except it is polled instead of interrupt driven. The Sequencer determines the state of the port by reading the XXFULL and IXFULL lines from the Status Input Port.

STATUS INPUT PORT:

The Status Input Port consists of two tri-state drivers that buffer the state lines of the interrupt sources as described above. It also allows the Sequencer to read the state of the FOOTSWITCH, SYNC and Analog to Digital Converter. The RC network of R19/20 and C42 provide some debouncing of the FOOTSWITCH.

CLICK OUTPUT:

The Click Output consists of Z17b, R36 pullup and bandpass filter R16/C35/R17. A 100pF capacitor at the Click Out Jack completes the filtering. Everytime the click address is accessed via the I/O Address Decoder, the output of Z17b toggles, generating a pulse at the Click Out Jack. Emphasis is placed on the first beat of the measure by outputting two pulses, 42 microseconds apart, for subsequent beats of the measure. Two pulses this close together reduce the low frequency energy and the pulse sounds lower in volume.

CLOCK CIRCUIT:

The Clock for the Sequencer can be either an internal clock of 1000 Hz or an external clock of any frequency between DC and 3000 Hz. The lower the frequency, the less recording/playing resolution available. Ideal input frequency is 1000 Hz whereas 24 Hz is barely acceptable and frequencies above 3000 Hz will keep the Sequencer in constant interrupt service. For more information about the external clock frequency, see APPENDIX F.

The flip/flop consisting of Z1c and Z1d allows switching between internal and external clocks by addressing the location decoded by the I/O Address Decoder.

The internal clock of 1000 Hz is generated by dividing the 1M Hz system clock by 1024 in Z19. The internal clock is disabled by pulling the reset line high, forcing the Q10 output to remain low.

The external clock circuitry consists of Z18b comparator circuit and associated components. This circuit senses when the input voltage crosses zero plus a .05V hysteresis level, at which time the output of Z18b saturates at the negative supply level. When the input falls below -.05V then Z18b switches back to the positive supply level. The circuit is enabled by a low level at the output of Z1c and creates an input voltage divider R47 and R42. When Z1c is high, the input of Z18b will always be high enough to keep it from switching and the output will be negative.

The output of Z18b is rectified, divided and filtered so that it is compatible with the clock input of Z17a (0V to 5V), which generates the interrupt. R48 is necessary to offset the effect of the nonsymmetrical load of Z18b. The interrupt is cleared by a pulse from the I/O Address Decoder when the Sequencer has determined, by reading the Status Input Port, that the interrupt came from the timer.

ANALOG TO DIGITAL CONVERTER:

The ADC circuit is centered around the National 0804 single channel IC. The circuit converts the resistance of a 100K linear potentiometer in the pedal housing to a digital value between 0 and 255.

The reference of the ADC is set by R25. This effectively adjusts the range of digital values obtained. The minus input for the ADC is set by R29. Adjustment of this trimmer sets the zero value with the pedal all the way up. The Z18a circuit converts the pedal resistance to a voltage source with low impedance, which is sent to the ADC for conversion.

Conversion ocurrs within 100 microseconds and the INTR line goes low, telling the Sequencer that a conversion is finished. When the Sequencer reads the ADC value, the RD line goes low and the digital data is sent to the Apple via the Z8 buffer. The ADC conversion process is started again by pulling the WR line low on the 0804, which is done by the pulse from the I/O Address Decoder after the Sequencer reads the value.

ADC ADJUSTMENT PROCEDURE

To adjust the ADC zero point and range, it is necessary to continuously view the value of the control pedal. You can do this by running USER UTILITY BANK 2, NUMBER 1 (ADC TEST) or USER UTILITY BANK 2, NUMBER 0 (the preferred Interface Test Program). Number 1 displays the value in decimal from 0 to 255 and NUMBER 0 will display the pedal value in HEX from 00 to FF when the **^A** command is issued (see APPENDIX L).

Once the Sequencer is continuously displaying, push the pedal all the way in the up position and adjust trimpot R29 (the one closest to the rear of the APPLE) for 00 display. You should first adjust R29 until you start seeing 01's then back off until it is always 00. Then push the pedal all the way down and adjust trimpot R25 (the one closest to you) for FF (or 255 decimal) display. You should adjust from FE until the display always reads FF. Then push the pedal all the way up again and readjust for 00. Exit the ADC Test mode by typing a <RET> and exit the Interface Test Program by typing **^E.**

SYNC, EXTERNAL CLOCK and CLICK SIGNAL SPECIFICATIONS

SYNC INPUT:

LEVEL : 0V to 5V, TTL (external device must be able to sink .9 mA @ .8V max, can use open collector because input is pulled up by 10K).

SPEED :Ton (min) = Toff (min) = 52 microseconds in SYNC CHECK mode (1.5 milliseconds in Single Step2 mode).

NOTE : If a footswitch is used for this input, it should be debounced unless sync delay is set to zero. Both single step timer sources debounce this input in software.

EXTERNAL CLOCK INPUT:

LEVEL : .7Vp-p minimum, 22Vp-p maximum.

INPUT IMPEDANCE : approximately 100K ohms.

WAVEFORM : Sine or Square (duty cycle 25% to 75%).

MINIMUM FREQUENCY : 0 Hz (although for useable resolution, a minimum of 48 Hz is recommended).

MAXIMUM FREQUENCY : depends on complexity of sequence but generally should be limited to 3000 Hz.

INTERNAL TIME INCREMENT : Adjustable from 1/16X to 8X clock frequency in powers of 2. This is used to match external clock as close as possible to internal clock if you want to switch clocks after you record the sequence.

CLICK OUTPUT:

OUTPUT LEVEL : -2V TO +2V nominal.

OUTPUT IMPEDANCE : 10K ohms.

NAME SYNTAX RULES

These rules apply to naming sequences, program groups, tracks and variable set-ups (from RECONFIGURE):

-15 characters maximum, not including the carriage return.

-First character must be a letter.

-Names must not include a comma.

-Control characters are not allowed (except ^U-right arrow, ^H-left arrow, ^X-cancel and ^M-return).

-The left and right arrow keys and ^X-cancel are allowed for editing the entry as long as the characters they produce meet these rules. ^X-cancel will sound a bell and clear the entry from the screen so you can start over.

-ESC editing sequences are not allowed.

-Leading spaces are not allowed.

-Trailing spaces are allowed and will count toward the 15 character maximum. You will not, however, need to type in trailing spaces when retreiving or deleting sequences or program groups.

The Sequencer will obtain each character and check to see that it obeys the above rules. If it does not, a bell will sound and that key will not appear on the screen or in the name . A RETURN for the first character will signify that you abort this entry, with implications depending on the circumstances detailed in this manual.

USING ASSEMBLY ROUTINES FROM ANOTHER BASIC SYSTEM

This Appendix shows you how to use some of the internal assembly routines for your own BASIC program. Included is a source listing of the PAGE 3.HEX variable table, which includes descriptions of the system variables accessible from BASIC.

CHROMA.BEGIN

To create a turn-key system you must first run CHROMA.BEGIN from drive 1. This is done by the HELLO program on dirve 1. CHROMA.BEGIN loads the assembly routines routines and initializes the system.

MODIFICATIONS TO CHROMA.BEGIN

CHROMA.BEGIN can be modified to display any prompting message by changing lines 140-170. Currently the message is:

```
**********************
*                    *
*  CHROMA SEQUENCER  *
*                    *
**********************

   COPYRIGHT 1982
     CBS INC.

PLEASE WAIT WHILE I GET SET UP...
```

Line 415 should be modified to run your program. Currently, it runs RECORD TRACK.1, which is the Sequencer control program.

In line 60, HIMEM: is set to 10219 ($27EB HEX), which is one location before the start of the sequence RAM. If you do not plan to keep a sequence resident in RAM for your application program, you can set HIMEM: to -27481 ($94A7 HEX). This will give you approximately 22K more RAM for your BASIC program.

YOUR APPLICATION PROGRAM

The first thing your program must do is set HIMEM: to the same value that you set in CHROMA.BEGIN. You must then declare the string variable N$ so that the assembly routines know where the variable resides. N$ is a variable used for file and track names.

You must then set BB% (or any other integer variable name) to 808. You should then call REINIT.CHROMA.PORT and REINIT.EXPANDER.PORT as described below.

When exiting your program you must POKE 1008,89: POKE 1009,250: POKE 1022,101: POKE 1023,255. This sets the break vector back to $FA59 and the interrupt request vector back to $FF65, the monitor entry point. It is also a good idea to set the hardware up for an external clock before you do this (see APPENDIX N).

CALLING AN ASSEMBLY PROGRAM

Before calling an assembly program, the CALNUM (location 877) must be set to a value representing the particular routine. CALNUM need not be reset for each successive call if you are calling the same routine. The data transfer location is decimal 6. Routines that input from or output to the ports or store or get data from the sequence queue use this location for the data byte. Below is a description and the CALNUM of each routine that can be directly called from BASIC:

FPTOSP CALNUM=0

This routine sets the sequence input and output pointers to the values stored in the sequence file. You will probably not use this unless you are manipulating a sequence.

LOGO.AND.CATALOG CALNUM=1

This routine displays a disk catalog as described in Chapter 7.

STOREO CALNUM=2

This routine stores a byte in the sequence queue at the current SIPTR (location 24,25) and updates SIPTR. The source is transfer location 6. If the sequence is full, ERRCOD (location 797) is set to 17, otherwise ERRCOD is not modified. In other words, if you want to check for a sequence full condition, you must set ERRCOD to 0, then check it after calling STOREO. It is left up to the programmer to recycle the queue and set the file pointers to SIPTR and SOPTR before saving the sequence.

GET2 CALNUM=3

This routine increments the SOPTR (location 26,27) if SOURCE (location 793) is zero (seq queue) or the command buffer BUFPNT if SOURCE is non-zero (command buffer). The command buffer is an 8 byte buffer that is not directly addressable from BASIC.

You can use this routine to pull data out of the sequence RAM. To do this, you would first set an integer variable (SO% for example) to the value in SOPTR. Then PEEK (SO%) to get the data. Then call this routine to point to the next byte. It is left up to the programmer to recycle the queue and set the file pointers to SIPTR and SOPTR before saving the sequence.

REINIT.CHROMA.PORT CALNUM=4

This routine reinitializes the Sequencer and the Chroma port. It also interrogates the instrument attached for ID and software revision information. It updates IDNUM (804), REVNUM (806), and sets MSKTBL (885) to zero, signifying one Chroma on line. Upon return, the following ERRCODs are possible:

00 - no error
01 - Chroma not responding to 260 NOPs
03 - Chroma not responding to input request
05 - Chroma not responding to output, the queue is full
07 - Chroma not echoing ID command

SEARCH.Q CALNUM=5

This routine loads the search (TRACK.DIR.HEX) overlay, if it is not in RAM, and calls the SEARCH.OPERATION routine, which performs functions depending on the value of SRCTYP, the search type number. SRCTYP can have any one of the following values:

VALUE	NAME	FUNCTION
00 -	FNDVOL:	cycle through the sequence and put the initial volume values in the Volume Table (starting at $94F5)
01 -	DELTRK:	delete a track then delete all time commands that are followed by another time command
02 -	CHGVOL:	change the initial volume of a track
03 -	TRANS1:	transpose a track or the entire sequence

04 - DELTM: delete time commands that are followed by another time command

05 - CHRNTM: quantizes all time values to within 6 time values

06 - DEMEAS: delete all measure commands, clear any click track information, and set the endpoints to beginning and ending.

07 - MUTE.UNMUTE: allows muting and unmuting of tracks by calling the MUTE/UNMUTE routine

08 - CHANGE.PORTS: allows the user to change the port of a track from Chroma to Expander or vice versa

09 - CHANGE.PROGRAM: allows the user to change the program number of a track

10 - CHANGE.NAME: allows the user to change the name of a track

PREREC CALNUM=6

This is the record/play routine. Several things have to be determined and/or set up prior to calling this routine. See the following lines in RECORD TRACK.1 for an example:

22-25, 100-153, 163-165

To Play, see lines 230, 500-2150
To Play Along, see lines 233, 500-2150
To Record, see lines 225, 500-2150

A full explanation of the set up required is beyond the scope of this APPENDIX. The information, however, will be available in the PROGRAMMERS MANUAL.

MONTOR CALNUM=7

This routine advances the sequence queue until a command block is found that contains the measure command for the measure number in DELMES (location 881). If the measure command is not found, ERRCOD is set to 18, otherwise ERRCOD is set to zero (no error). Upon entry, SOPTR must point to the first time command in the sequence queue after the EOS command. When advancing the queue, all DEFINE, UNDEFINE and performance commands are outputted to the ports. ATTACK and RELEASE commands are not outputted. It is left up to the programmer to recycle the queue and set the file pointers to SIPTR and SOPTR before saving the sequence.

OUT1 CALNUM=8

This is the general routine that outputs a byte in transfer location 6 to the Chroma port. If the output queue is empty and the Chroma is not busy then this routine will output directly to the Chroma. Otherwise, it will store the byte in the output queue. If the output queue is full, this routine will keep trying 255 times. The routine will then return with the carry cleared if successful or the carry set if unsuccessful. Unfortunately, there is no way from BASIC for you to know if the data was transferred, since you can not access the carry flag. You could write a small assembly routine that calls this routine and checks the carry flag, setting ERRCOD appropriately. For example:

```
OUTPUT LDA #$00
       STA ERRCOD    CLEAR ERROR CODE
       LDA #$08
       STA CALNUM    SET UP FOR OUT1
       JSR 808       CALL OUT1
       BCC NOERR     ALL IS WELL...
       LDA #$05      CHROMA NOT RESPONDING TO OUTPUT CODE
       STA ERRCOD    SET ERROR CODE REGISTER
NOERR  RTS           RETURN FROM OUTPUT
```

INPUT CALNUM=9

This is the general routine to input from the Chroma port. The routine will wait indefinitely for a byte from the Chroma and return after storing the byte in transfer location 6.

CLICK.SET.UP CALNUM=10

This loads the click set up overlay (CLICK.HEX) if it is not already in RAM, which allows the user to set up and change the click track parameters as described in Chapter 8.

EDITOR CALNUM=11

This loads the editor overlay (EDITOR.HEX) if it is not already in RAM, which allows the user to perform several editing functions on the sequence as described in Chapter 6.

RESTORE CALNUM=12

This routine restores the Chroma instruments according to the program in the LED display. This involves undefining instruments 2-7 and redefining 0 and 1 according to the program. The panel, performance and pressure switches are also turned off. A fatal error (returns back to BASIC without resetting sequence pointers) if the Chroma is not communicating properly (SYNC ERROR).

TRANSFER CALNUM=13

This loads the transfer overlay (TRANSFER.HEX) if it is not already in RAM, which allows direct communication between the Chroma and Expander as described in Chapter 10.

SPTOFP CALNUM=14

This routine sets the sequence file input and output pointers to the values stored in SOPTR and SIPTR. If you move the sequence in any way, you must cycle through the queue until the EOS, then call this routine before you save the sequence. Unless specifically stated, all routines accessable from BASIC take care of SOPTR, SIPTR and the file pointers for you.

SAVE.SEQUENCE CALNUM=15

This is the routine that saves a sequence as described in Chapter 7.

AUTO.SAVE.ENTRY CALNUM=16

This is the routine that performs the AUTOSAVE function as described in Chapter 5.

PROG.FILE.MANAGE CALNUM=17

This is the routine that handles the transfer of program groups from the Chroma and Expander to disk or vice versa as described in Chapter 7.

GET.SEQUENCE CALNUM=18

This is the routine that loads a sequence file from disk as described in Chapter 7.

CLEAR.SEQUENCE.BASIC CALNUM=19

This is the routine that clears a sequence from RAM as described in Chapter 5.

PRINT.LAST.TIME CALNUM=20

This routine prints "LAST TIME: XXXXX" where XXXXX is the decimal value of the register LASTM, which contains the value of the last time command encountered during playback or record.

DELETE.SEQUENCE CALNUM=21

This routine allows the user to delete a sequence file from disk as described in Chapter 7.

METRO.SET.UP CALNUM=22

This routine sets up click track parameters to NONE, 60 BPM and 4/4 time signature.

RECONFIGURE CALNUM=23

This routine loads the reconfigure overlay (SYS.PORT.HEX), if it is not already in RAM, which allows the user to change various operating parameters of the Sequencer as described in Chapter 9.

REINIT.EXPANDER.PORT CALNUM=24

Same as REINIT.CHROMO.PORT but operates on the Expander port. MSKTBL is set to 01 if an Expander is responding. The ERRCOD values returned are described below:

00 - no error
02 - Expander not responding to 260 NOPs
04 - Expander not responding to input request
06 - Expander not responding to output
08 - Expander not echoing ID command

MARKER CALNUM=25

This routine loads an overlay called MARKER.HEX which does not exist at this time. It is reserved for a program that will allow the user to set markers commands in the sequence for use by the play routine and the Editor.

CVOLUME CALNUM=26

This routine loads an overlay called VOLUME.HEX which does not exist at this time. It is reserved for a program that will allow the user to vary the volume of a track continuously by using the control pedal.

MENU.PAGE.3 CALNUM=27

This routine loads the menu overlay (MENU3.HEX), if not already in RAM, which will clear the screen and display page three of the main menu.

TRACK.DIR CALNUM=28

This routine loads the search overlay (TRACK.DIR.HEX), if not already in RAM and calls the Track Directory routine within that overlay. This will clear the screen and display the current track directory.

USERA CALNUM=29

This routine will load and run the USER UTILITY BANK 1 program. Upon entry, you must set USERNO (location 872) to the bank 1 user number as described in APPENDIX C.

USERB CALNUM=30

Same as USERA but will load and run a USER UTILITY BANK 2 program.

GET.NAME.NB CALNUM=31

This routine gets a name from the user, applying the name syntax rules as described in APPENDIX P. The name is then transferred to the Applesoft variable N%.

ACCESS TO OTHER ASSEMBLY ROUTINES

Although there is no direct access to the other assembly routines from BASIC, a small assembly program can be written to access any byte in the Sequencer. Essentially all that is required of such a program is to switch the RAM card to RAM BANK 2 read/write and JMP to the location. Upon return from the desired routine, the small program would reset the RAM card to ROM read and return to your BASIC program. 4096 bytes of ASCII messages and print routines reside in the RAM card bank 1. The locations and descriptions of all the routines in the Sequencer are currently available only in the soon to be published PROGRAMMERS MANUAL.

INTERRUPTS

The Sequencer card should not be used with hardware and/or software that generates or uses interrupts (IRQ or BRK) or software that inadvertently enables interrupt requests. This also applies to utility programs that generate BASIC or assembly programs that run on the system. See APPENDIX N for an explanation and possible solutions to this kind of problem. It is also possible to change the interrupt vectors to a routine that just does a LDA $45 and a RTI for systems that inadvertently enable the interrupts.

PAGE THREE

The Sequencer assembly routines use all of page three ($300-$3CF) for a variable table and RAM switching routines. Your BASIC or assembly routines must not use this space for anything other than controlling the Sequencer routines as prescribed in this APPENDIX and the Programmer's Manual. It is possible to move the page three table to another location, but this would require alteration of CHROMA.BEGIN, RECORD TRACK.1 (if used) and several file management and RECONFIGURE routines.

ZERO PAGE

There are many zero page locations that are used by the Sequencer assembly routines. Locations used exclusively by the Sequencer are $06-09 and $18-1F. Other locations are used, but these are used with a knowledge of their effects on the Monitor and BASIC interpreter. In other words, it is something that you normally will not care about. If your program (or any utility program you use to write your program) uses $06-09 or $18-1F, you will have to save the data in these locations in both directions (your program to the Sequencer assembly programs and vice versa). Locations $07 and $18-1F are initially set up in CHROMA.BEGIN.

Also, it is important to insure that interrupts are not enabled or a BRK instruction is not executed in your program. You could execute a BRK if you change the BRK vector, but you must make sure that you restore it before calling any of the Sequencer's assembly routines.

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
9	RSC 432		
10	RSC 433		
11	RSC 434		
12	RSC 461		

EXPANDER PORT

CHROMA PORT

I/O ADDRESS DECODER

CLICK OUTPUT

INTERRUPT CONTROL

ANALOG TO DIGITAL CONVERTER

CLOCK CIRCUIT

STATUS INPUT PORT

POWER

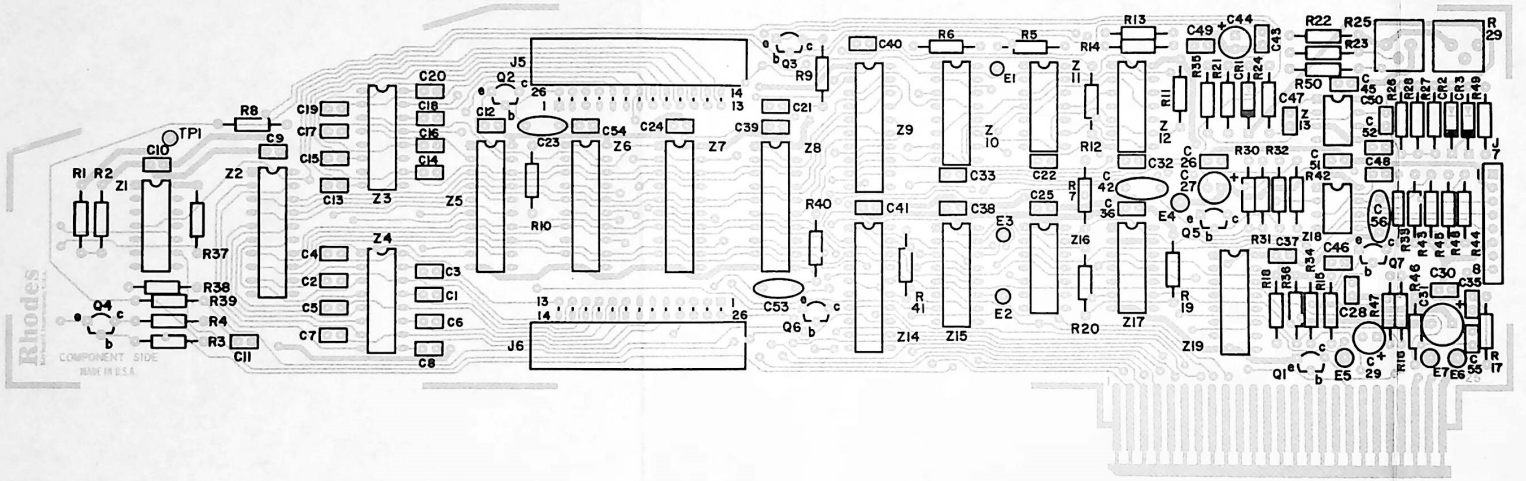
NOTES:

1. UNLESS OTHERWISE SPECIFIED:
   ALL RESISTOR VALUES ARE IN OHMS; ¼ W, 5%
   ALL CAPACITOR VALUES ARE IN μF, (P= PICOFARADS)
2. HIGHEST REFERENCE DESIGNATIONS: Z19, C58, R50, CR4, Q7, J7, E10
3. J5-5 & 7, J6-5 & 7 NOT USED.
4. CONVENTIONS USED FOR SUPPLY VOLTAGE CONNECTIONS:

↑	IMPLIES +12V ANALOG
↓	IMPLIES -12V ANALOG
⏚	IMPLIES ANALOG GROUND
△	IMPLIES +5V DIGITAL
▽	IMPLIES DIGITAL GROUND
⏚	IMPLIES CHASSIS GROUND

ARP PART NO	UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES	TITLE	SCHEMATIC, PC BD, CHROMA/COMPUTER INTERFACE
USED ON PL30-7226601	TOLERANCES XX ± .03, XXX ± .010, ANGLES ± 1°	SIZE	D
MATERIAL	REMOVE BURRS & SHARP EDGES, DO NOT SCALE DRAWING	DRAWING NO	30-77241
FINISH	DRAWN REM 10/6/82	REV	12
	CHECKED	SCALE	NONE
	APPROVED	SHEET	1 OF 1
	APPROVED		

REDUCE TO 12.000 ±.005



FENDER, RHODES & ROGERS			
COMP / CHROMA INTERFACE			
SIZE D	FSCM NO.	DWG. NO. 30-45587	REV. —
SCALE 2/1		SHEET	OF